commodore
COMPUTER CLUB
N. 7
Lire 2000
Mensile per gli utenti di sistemi Commodore
Dicembre 1983 - Sped. Abb. Post. gr.III/70 - Suppl. a Computer n 62 - Distr. Me. Pe. Spa
Due regali eccezionali:
Un assembler
per tutti i Commodore
e la maschera - memo
del programmatore
Giochi e programmi
per il VIC
ed il 64

espande all'infinito la tua esperienza

software a misura d'uomo

SOMMARIO

Commodore Computer Club - mensile indipendente per gli utenti di sistemi Commodore
Direttore responsabile: Michele di Pisa
Redazione: Alessandro De Simone
Direzione, redazione: V.le Famagosta, 75 - 20142 Milano - Tel. 02 / 8467348
Pubblicità: Milano - Micro Croce, Paola Bevilacqua, Gianluigi Centurelli, Tina Ronchetti, Villa Claudio - V.le Famagosta, 75 - 20142 Milano - Tel. 02 / 8467348/9/40
Prezzi e abbonamenti: Prezzo per una copia Lire 2.000
Arretrati il doppio. Abbonamento per dieci fascicoli lire 18.000
Abbonamento annuo cumulativo alle riviste Computer e Commodore Computer Club (tariffa riservata agli studenti) L. 34.000 . I versamenti vanno indirizzati a Minisystem - Italia s.r.l., mediante assegno bancario, vaglia o utilizzando il c/c postale n. 11909207.
Composizioni: Minisystem - Italia
Selezioni: Org. A. G.
Stampa: La Litografica s.r.l. - Busto A.
Registrazione: Tribunale di Milano n. 2/10/1982 - Sped. in abb. post. gr. III n. 70 quale supplemento alla rivista Computer - Pubb. inferiore al 70%

Con i nuovi programmi italiani della Sirius Elettronica, Commodore 64 serve proprio a tutto.

Per tutti i possessori, anche futuri, del personal computer Commodore 64 c'è un'importante novità: la creazione di una vasta serie di nuovi programmi realizzati espressamente per le esigenze italiane.

Con questi programmi i tecnici italiani della Sirius Elettronica mettono a disposizione un servizio che trova Utenti nei più diversi settori delle attività produttive: dal dentista all'albergo, dal negozio al teatro. E molti altri ancora.

Programmi semplici da usare, chiari, e razionali. E in italiano.

Tutto questo per valorizzare sempre di più Commodore 64.

Desidero conoscere il punto vendita, a me più vicino, in cui ottenere informazioni sui programmi applicativi della Sirius Elettronica.

Nome ____________

Cognome ____________

Via ____________

CAP ______ Città ____________

Inviare a:
Sirius Elettronica
Via Imperia, 21 - 20142 Milano
Tel. 02/8490400-8431368

PROGRAMMI SU DISCO PER CBM - 64

	Prezzo al pubblico (IVA esclusa)
GESTIONE AGENTI RAPPRESENTANTI	120.000
GESTIONE AMMORTAMENTI MUTUI	60.000
GESTIONE ANAGRAFICHE	120.000
GESTIONE SCADENZE POLIZZE	300.000
GESTIONE AGENZIA VIAGGI	250.000
TOTOCALCIO (SVILUPPO COLONNARE)	100.000
GESTIONE CONTI CASA	100.000
GESTIONE CONDOMINIO (SEMPLIFICATO)	200.000
IMPARIAMO IL BASIC	100.000
DICHIARAZIONE I.V.A.	150.000
DISTINTA BASE	80.000
GESTIONE EQUO CANONE	100.000
MODELLO 740 ORDINARIO	100.000
MODELLO 740 NORMALE	150.000
GESTIONE STAMPA FATTURA	80.000
CENTO PROGRAMMI PER CBM-64 (DIDATTICO)	80.000
GESTIONE FIDO CLIENTI	120.000
GESTIONE CONTO CORRENTE	150.000
GESTIONE CONTI	150.000
GESTIONE APPUNTAMENTI	150.000
GESTIONE ORDINI	150.000
GESTIONE CANTINE	250.000
GESTIONE I.V.A. SEMPLIFICATA	60.000
GESTIONE CLUBS	250.000
GESTIONE OFFICINE	250.000
GESTIONE PORTAFOGLIO CLI/FOR	200.000
GESTIONE TEATRO	120.000
GESTIONE LIBRERIE E BIBLIOTECHE	150.000
GESTIONE RISTORANTI	380.000
MAILING LIST	150.000
WORD PROCESSOR	160.000
WORD ON MAIL	250.000
SCADENZIARIO EFFETTI CLI/FOR	200.000
GESTIONE PULIZIA CAPI	200.000
CONTABILITÀ FATTURE C/F	200.000
CONTABILITÀ MONOAZIENDALE SEMPLIFICATA	400.000
GESTIONE ALBERGHI	400.000
GESTIONE PARRUCCHIERI	400.000
GESTIONE GOMMISTI	400.000
LEGGE 373 (ISOLAMENTI TERMICI)	150.000
COMPUTO METRICO	400.000
INGEGNERIA CIVILE I	250.000
INGEGNERIA CIVILE II	250.000
REVISIONE PREZZI	300.000
MAGAZZINO E FATTURAZ. AGGANCIATE	400.000
CONTABILITÀ GENERALE	400.000
GESTIONE OTTICI	300.000
GESTIONE DENTISTI	300.000
GESTIONE FARMACIE	400.000
GESTIONE STUDI MEDICI	300.000
GESTIONE CONDOMINIO ADVANCED	400.000
DUPLICATORE ARCHIVI	400.000
SINTETIZZATORE MUSICALE	120.000
DATA BASE	200.000

PROGRAMMA APPLICATIVI SU CASSETTA PER VIC-20

SIR-MAG. 8	62.500
SIR-MAG. 16	87.000
SIR-MAG. 32	98.000
RICETTARIO DI CUCINA	38.000
BIORITMO	41.000
PROGRAMMA PUBBLICITARIO PER NEGOZI	41.000
INDIRIZZARIO	41.000
SIR-CONT - CONTABILITÀ DOMESTICA	49.000
PROGRAMMI GIOCO SU CASSETTA	24.500
GORTEK AND MICROCHIPS	24.500

Può un computer imparare dai propri sbagli?

Un piccolo esempio di intelligenza artificiale.

QUESTO che presentiamo è uno dei programmi che fanno molto riflettere sulle possibilità di un calcolatore. Si dimostra, infatti, che, attraverso una serie di domande e risposte, un computer può "apprendere" informazioni e fornirle quando si richiedono.

Una volta che il lettore digita il programma (che gira su qualsiasi computer Commodore, compreso il Vic 20 senza espansioni), può provarne il funzionamento, rispettando la seguente procedura riportata a solo titolo di esempio (in maiuscolo ciò che il lettore deve digitare, in minuscolo i messaggi del computer).

Digita il primo nome della serie

PASQUALE

e quando sei pronto premi un tasto

Premere un tasto qualunque.

Pasquale. Ho indovinato (s/n)?

N

Che cos'era?

SANDRO

Che domanda faresti parlando di Sandro?

PORTA GLI OCCHIALI

Che risposta daresti (s/n)?

S

Vuoi giocare ancora (s/n)?

S

Pensa a ciò che devo indovinare... e quando sei pronto premi un tasto

Porta gli occhiali?

N

Pasquale. Ho indovinato (s/n)?

N

Cos'era?

ANNA

Che domanda faresti parlando di Anna?

E' intelligente

Che risposta daresti (s/n)?

N

Eccetera. Come si può notare il computer, all'inizio, ha a disposizione solo un nome (Pasquale) e chiede subito se il nominativo da indovinare è proprio Pasquale oppure no. In caso negativo gli si comunica il nuovo nome (Anna), che verrà memorizzato insieme alla sua caratteristica peculiare. All'inizio il gioco è monotono, ma diventa divertente dopo l'inserimento di una ventina di nominativi, che possono anche esser ripetuti inserendo ogni volta una caratteristica diversa. Per esempio, dapprima Pasquale sarà solo

```
10 REM *** APPRENDIMENTO AUTOMATICO PER COMPUTER ***
20 REM ***                                       ***
30 REM *** PER QUALSIASI COMPUTER (ALMENO CON 5K)***
40 REM ***                                       ***
50 REM ***PASQUALE D'AURIA VIA E. DE FILIPPIS,145***
60 REM ***        84013 CAVA DEI TIRRENI (SA)      ***
70 REM ***       TEL. 089/462617                   ***
80 REM ***                                       ***
100 DIM A$(300), A%(150): X$ = "(S/N)"
110 PRINT "LEGGO DA NASTRO?" X$
120 GOSUB 630 :IF B$ = "" OR B$ <> "S" AND B$ <> "N" THEN 120
130 IF B$ = "N" THEN 610
140 OPEN 1,1,0
150 J=0
160 INPUT#1,A$(J):INPUT#1,A%(J)
170 IF A$(J) <> "FINE" THEN J=J+1: GOTO 160
180 CLOSE 1
190 I=0: PRINT "PENSA A CIO' CHE DEVO INDOVINARE..."
200 PRINT "...E QUANDO SEI PRONTO PREMI UN TASTO.": GOSUB 630
210 IF ASC (A$(I)) = 64 THEN 270
220 PRINT A$(I) "?"
```

SE VUOI ESSE
DI SCEGLIERE

Ogni giorno in Europa si apre un nuovo computer shop.
Un pubblico sempre più numeroso è attirato verso il personal
e si rivolge ai negozi specializzati per trovare la sua marca
preferita.
Tu che hai capito qual è il futuro dei computer e hai deciso
di aprire un negozio,
cerchi un nome che dia prestigio
e una organizzazione che
non ponga vincoli ma offra
vantaggi concreti.
Computeria vuol dire negozi
di computer fin dal 1979.
Computeria è anche una
organizzazione che ha avviato
rapporti di collaborazione
con tutti i principali fornitori,
perciò i suoi affiliati possono
scegliere e vendere le marche
più prestigiose e richieste.
E inoltre Computeria ti dà un prezioso
know-how, una ricchissima dotazione di
programmi, supersconti esclusivi,
vantaggi economici sul leasing.
E tanta pubblicità.
Se vuoi essere libero di scegliere quello che
vuoi vendere nel tuo negozio,
l'organizzazione Computeria
è la tua scelta obbligata.

COMPUTERIA®

La catena senza catene.

RE LIBERO

occhialuto, in sequito sarà elegante e così via.

Com'è intuitivo, la possibilità di memorizzare una gran quantità di nominativi o di servizi di lunghe frasi esplicative, dipende dalla memoria disponibile. Nel caso si disponga di un consistente numero di byte (come nel Commodore 64) si può modificare la linea 100.

Quando si desidera smettere verrà posta la domanda se registrare l'archivio oppure no. In caso affermativo si preparerà un nastro cassetta e si seguirà la nota procedura di registrazione. Chi possiede l'unità a dischi non dovrebbe avere difficoltà a modificare la routine di lettura e scrittura dei dati. ▪

```
230 GOSUB 630
240 IF B$= "S" THEN I=A%(I): GOTO 210
250 IF B$<> "N" THEN 230
260 I=A%(I)+1: GOTO 210
270 PRINT "▒" RIGHT$(A$(I), LEN(A$(I))-1)
280 PRINT "HO INDOVINATO?" X$
290 GOSUB 630
300 IF B$= "S" THEN 460
310 IF B$ <> "N" THEN 290
320 IF FRE(0) < 100 THEN PRINT "HAI A DISPOSIZIONE ": PRINT FRE(0) "CARATTERI"
330 INPUT "CHE COS'ERA"; B$
340 PRINT "CHE DOMANDA FARESTI PARLANDO DI "B$: INPUT C$
350 PRINT "CHE RISPOSTA DARESTI  " X$:INPUT D$
360 IF D$ <> "S" THEN 400
370 A$(J) = "@" + B$
380 A$(J+1) = A$(I)
390 GOTO 430
400 IF D$ <> "N" THEN 350
410 A$(J+1) = "@" + B$
420 A$(J) = A$(I)
430 A$(I) = C$
440 A%(I) = J
450 J = J+2: A$(J) = "FINE"
460 PRINT "VUOI GIOCARE ANCORA?"
470 GOSUB 630
480 IF B$ = "S" THEN 190
490 IF B$ <> "N" THEN 470
500 PRINT "DEVO REGISTRARE?"
510 GOSUB 630
520 IF B$ = "N" THEN END
530 IF B$ <> "S" THEN 510
540 OPEN 1,1,1
550 I=0
560 PRINT#1,A$(I)
570 PRINT#1,A%(I)
580 I=I+1
590 IF I <= J THEN 560
600 CLOSE 1
610 PRINT "▒DIGITA IL PRIMO NOME  DELLA SERIE": INPUT B$
620 A$(0) = "@" + B$:J=1: GOTO 190
630 GET B$: IF B$ = "" THEN 630
640 RETURN
```

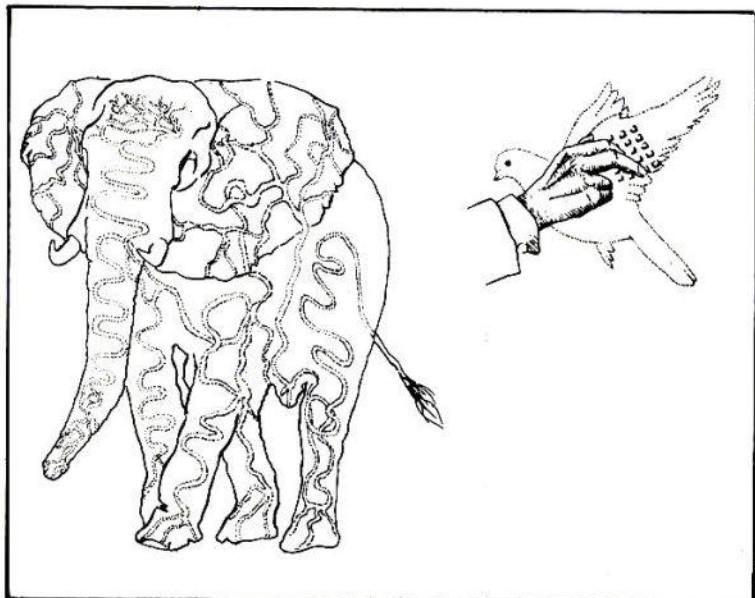
Un programma assemblatore

Digitare questo programma non è semplice nè molto divertente. Alla fine, però, scoprirete un nuovo modo di programmare.

IL programma assemblatore inizia alla riga 1111, perchè nelle prime 1111 linee (dalla 0 alla 1110), andrà scritto il programma da assemblare. Il formato di una riga dovrà essere:
'numero, virgolette, istruzione in assembler.

Esempio: 10 " lda #234

Vi è la possibilità di inserire dei commenti che dovranno essere preceduti dal punto e virgola (;). Si dovrà dunque scrivere:
10 " lda #234; commento qualunque
oppure:
10 "; commento qualunque

E' necessario, però, ricordare che la prima istruzione di qualsiasi programma in assembler, dovrà essere del tipo:
10 "* #xxxx
ove al gruppo xxxx deve essere sostituito il valore (nel caso nostro decimale) di partenza di assemblaggio. In altri termini (se si vuole che la nostra routine in l.m. cominci dalla locazione esadecimale 033c, bisognerà scrivere:
10 "*$033c ; punto start
il riferimento ad etichette mediante l'uso del prefisso @ (chiocciola), seguito dal nome della label.

Istruzioni del microprocessore 6502

```
116 ";  SOMMA CON   124 "  ADC #MI      134 "  ADC .X(MI)   144 "  AND .(MI)
118 ";  RIPORTO     126 "  ADC ,X(MA)   136 ";              146 "  AND #MI
120 "  ADC (MA)     128 "  ADC ,Y(MA)   138 ";AND LOGICO    148 "  AND ,X(MA)
122 "  ADC .(MI)    130 "  ADC /X(MI)   140 ";              150 "  AND ,Y(MA)
                    132 "  ADC /Y(MI)   142 "  AND (MA)     152 "  AND /X(MI)
```

```
154 "   AND /Y(MI)
156 "   AND .X(MI)
158 ";CONFRONTA
160 ";CON
162 "ACCUMULATORE
164 "   CMP (MA)
166 "   CMP .(MI)
168 "   CMP #MI
170 "   CMP ,X(MA)
172 "   CMP ,Y(MA)
174 "   CMP /X(MI)
176 "   CMP /Y(MI)
178 "   CMP .X(MI)
180 ";
182 "OR ESCLUSIVO
184 ";
186 "   EOR (MA)
188 "   EOR .(MI)
190 "   EOR #MI
192 "   EOR ,X(MA)
194 "   EOR ,Y(MA)
196 "   EOR /X(MI)
198 "   EOR /Y(MI)
200 "   EOR .X(MI)
202 ";CARICA L'
204 ";
206 "ACCUMULATORE
208 "   LDA (MA)
210 "   LDA .(MI)
212 "   LDA #MI
214 "   LDA ,X(MA)
216 "   LDA ,Y(MA)
218 "   LDA /X(MI)
220 "   LDA /Y(MI)
222 "   LDA .X(MI)
224 ";
226 ";OR LOGICO
228 ";
230 "   ORA (MA)
232 "   ORA .(MI)
234 "   ORA #MI
236 "   ORA ,X(MA)
238 "   ORA ,Y(MA)
240 "   ORA /X(MI)
242 "   ORA /Y(MI)
244 "   ORA .X(MI)
246 ";SOTTRAE CON
248 ";
250 ";RIPORTO
252 "   SBC (MA)
254 "   SBC .(MI)
256 "   SBC #MI
258 "   SBC ,X(MA)
260 "   SBC ,Y(MA)
262 "   SBC /X(MI)
264 "   SBC /Y(MI)
266 "   SBC .X(MI)
268 ";IMMAGAZZINA
270 "; L'
272 "ACCUMULATORE
274 "   STA (MA)
276 "   STA .(MI)
278 "   STA ,X(MA)
280 "   STA ,Y(MA)
282 "   STA /X(MI)
284 "   STA /Y(MI)
286 "   STA .X(MI)
288 ";SPOSTAMENTO
290 ";ARITMETICO
292 ";A SINISTRA
294 "   ASL A
296 "   ASL (MA)
298 "   ASL .(MI)
300 "   ASL ,X(MA)
302 "   ASL .X(MI)
304 ";SPOSTAMENTO
306 ";LOGICO
308 ";A DESTRA
310 "   LSR A
312 "   LSR (MA)
314 "   LSR .(MI)
316 "   LSR ,X(MA)
318 "   LSR .X(MI)
320 "; ROTAZIONE
322 ";
324 "; A DESTRA
326 "   ROR A
328 "   ROR (MA)
330 "   ROR .(MI)
332 "   ROR ,X(MA)
334 "   ROR .X(MI)
336 "; ROTAZIONE
338 ";
340 "; A SINISTRA
342 "   ROL A
344 "   ROL (MA)
346 "   ROL .(MI)
348 "   ROL ,X(MA)
350 "   ROL .X(MI)
352 ";
354 "@UNO
356 ";
358 ";DIRAMAZIONI
360 ";
362 "   BCC UNO
364 "   BCS UNO
366 "   BEQ UNO
368 "   BMI UNO
370 "   BNE UNO
372 "   BPL UNO
374 "   BVC UNO
376 "   BVS UNO
378 ";
380 ";IMPLICATI
382 ";
384 "   BRK
386 "   CLD
388 "   CLC
390 "   CLI
392 "   CLV
394 "   DEX
396 "   DEY
398 "   INX
400 "   INY
402 "   NOP
404 "   PHA
406 "   PHP
408 "   PLA
410 "   PLP
412 "   RTI
414 "   RTS
416 "   SEC
418 "   SED
420 "   SEI
422 "   TAX
424 "   TAY
426 "   TSX
428 "   TXA
430 "   TXS
432 "   TYA
434 ";DECREMENTA
436 ";
438 ";LA MEMORIA
440 "   DEC (MA)
442 "   DEC .(MI)
444 "   DEC ,X(MA)
446 "   DEC .X(MI)
448 ";INCREMENTA
450 ";
452 ";LA MEMORIA
454 "   INC (MA)
456 "   INC .(MI)
458 "   INC ,X(MA)
460 "   INC .X(MI)
462 ";
464 ";SALTA A
466 ";
468 "   JMP (MA)
470 "   JMP /(MI)
472 ";SALTA ALLA
474 ";
476 ";SUBROUTINE
478 "   JSR (MA)
480 ";
482 ";CARICA X
484 ";
486 "   LDX (MA)
488 "   LDX .(MI)
490 "   LDX #MI
492 "   LDX ,Y(MA)
494 "   LDX .Y(MI)
496 ";
498 ";CARICA Y
500 ";
502 "   LDY (MI)
504 "   LDY .(MI)
506 "   LDY #MI
508 "   LDY ,X(MA)
510 "   LDY .X(MI)
512 ";IMMAGAZZINA
514 ";
516 "; X
518 "   STX (MA)
520 "   STX .(MI)
522 "   STX .Y(MI)
524 ";IMMAGAZZINA
526 ";
528 "; Y
530 "   STY (MA)
532 "   STY .(MI)
534 "   STY .X(MI)
536 ";VERIFICA
538 ";
540 "; DI BIT
542 "   BIT (MA)
544 "   BIT .(MI)
546 ";CONFRONTA
548 ";
550 ";CON X
552 "   CPX (MA)
554 "   CPX .(MI)
556 "   CPX #MI
558 ";CONFRONTA
560 ";
562 ";CON Y
564 "   CPY (MA)
566 "   CPY .(MI)
568 "   CPY #MI
```

Il listato del programma

```
 1111 REM"────────────────────
 1112 REM"
 1113 REM"     DI VIZIO   MAURIZIO
 1114 REM"
 1115 REM"     TEL 039 / 743092  .
 1116 REM"
 1117 REM"
 1118 REM"   -- ASSEMBLATORE 6502 --
 1119 REM"────────────────────
 5000 CLOSE3:GOTO6840
 5001 CLR:INPUT"▣STAI USANDO IL PET ";SU$:IFLEFT$(SU$,1)="S"THENWU=41:GOTO5010
 5002 WU=44
 5010 DIML$(8),VL(8):PRINT"⊐"
 5020 IB=PEEK(WU)*256+PEEK(WU-1)-1:YY=IB
 5030 PRINT"▣"
 5040 GOTO5090
 5050 :
 5060 S$=""
 5070 IFPEEK(IB)=0THENRETURN
 5080 S$=S$+CHR$(PEEK(IB)):IB=IB+1:GOTO5070
 5090 :
 5100 PRINT"LINEA/PROG.COUNTER▣":PRINT
 5110 VOLTA=1
 5120 LA=PEEK(IB+4)*256+PEEK(IB+3):IFLA=1111THEN5490
 5130 PRINTLA,PC:IB=IB+5
 5140 GOSUB5050
 5150 M$=CHR$(34)+"*":IFVO=1ANDLEFT$(S$,2)<>M$THENPRINT"⊐DA DOVE ALLOCO ?","ERR.
IN";LA:END
 5160 IFLEFT$(S$,2)=M$THEN 6750
 5170 IFLEFT$(S$,1)<>CHR$(34)THENPRINT"⊐ERRORE IN";LA:END
 5180 IF MID$(S$,2,1)=";"THEN5120
 5190 IF MID$(S$,2,1)="@"THEN5410
 5200 IFMID$(S$,2,1)="="THEN6500
 5210 GOTO5300
 5220 FORK=1TOLEN(S$):IFMID$(S$,K,1)="'"THEN5240
 5230 NEXT:PRINT"⊐ERRORE IN";LA:END
 5240 S$=MID$(S$,K+1,LEN(S$))
 5250 FORK=1TOLEN(S$):IFMID$(S$,K,1)="'"THEN5280
 5260 IFK>80THENPRINT"⊐TESTO TROP. LUNGO IN";LA:END
 5270 NEXT:PRINT"⊐ERRORE IN";LA:END
 5280 S$=MID$(S$,1,K-1):PC=PC+LEN(S$):IFPR=0THEN5120
 5290 PR=0:FORK=1TOLEN(S$):H=ASC(MID$(S$,K,1)):GOSUB6630:PC=PC+1:NEXT:GOTO5570
 5300 FORK=1TOLEN(S$):IFMID$(S$,K,1)=" "ORMID$(S$,K,1)=CHR$(34)THENNEXTK
 5310 S$=MID$(S$,4+K,LEN(S$)-K+1)
 5320 IF LEFT$(S$,1)=" "ORLEFT$(S$,1)=";"ORLEFT$(S$,1)=""THENPC=PC+1:GOTO5120
 5330 R$=MID$(S$,2,1):IFLEFT$(S$,1)="A"AND(R$=" "ORR$=";"ORR$="")THENPC=PC+1:GOT
O5120
 5331 SA$=LEFT$(S$,1)
 5340 IFSA$="#"ORSA$="$"THENPC=PC+2:GOTO5120
 5350 IFSA$="("THENPC=PC+3:GOTO5120
 5360 IFSA$=","THENPC=PC+3:GOTO5120
```

```
5370 IFSA$="."THENPC=PC+2:GOTO5120
5380 IFSA$="="THENIFMIS$(S$,2,1)="("THENPC=PC+3:GOTO5120
5390 IFSA$="="THENPC=PC+2:GOTO5120
5400 PC=PC+2:GOTO5120
5410 :
5420 S$=RIGHT$(S$,LEN(S$)-2):FORK=1TOLEN(S$):G$=MID$(S$,K,1):IFG$=" "ORG$=";"TH
EN5440
5430 NEXT:GOTO5450
5440 S$=MID$(S$,1,K-1)
5450 FORN=0TOI:IFL$(N)=S$THENPRINT"]LAB. RIPET.ERRORE IN";LA:END
5460 NEXT:I=I+1
5470 L$(I)=S$:VL(I)=PC
5480 GOTO5120:
5490 :
5500 PRINT"■=====================■":PRINT
5510 PRINT"■ETICHETTA/","VALORE■":PRINT
5520 PRINT:FORN=1TOI:PRINTL$(N),"="VL(N):NEXT:PRINT
5530 PRINT"■=====================■"
5540 FORBL=1TO2500:NEXT:PRINT"]■■"
5550 PRINT"]*** CODICE OGGETTO ***"
5560 FF=1:IB=YY:PC=0:TP=PC-1
5570 S$="":LA=PEEK(IB+4)*256+PEEK(IB+3):IFLA=1111THEN6830
5580 IB=IB+5
5590 GOSUB5050
5600 IFLEFT$(S$,1)<>CHR$(34)THENPRINT"]ERRORE IN";LA:END
5610 IF MID$(S$,2,1)=";"THEN5570
5620 IF MID$(S$,2,1)="="THEN5570
5630 IF MID$(S$,2,1)="@"THEN5570
5640 IF MID$(S$,2,1)="*"THEN:::IN=1:GOTO5150
5650 S$=MID$(S$,2,LEN(S$)):FORK=1TOLEN(S$):IFMID$(S$,K,1)=" "ORMID$(S$,K,1)=":"
THENNEXTK
5660 SA$=MID$(S$,K,3)
5670 SP$=MID$(S$,K+4,LEN(S$))
5680 FORK=1TOLEN(SP$):IFMID$(SP$,K,1)=";"ORMID$(SP$,K,1)=" "THENK=LEN(SP$):GOTO
5700
5690 NEXT:GOTO5710
5700 SP$=MID$(SP$,1,K-1)
5710 FORK=1TOLEN(SP$):G$=MID$(SP$,K,1)
5720 IFG$=" "ORG$=")"ORG$=";"THEN5740
5730 NEXT:GOTO5750
5740 SP$=MID$(SP$,1,K)
5750 :
5760 ST$=LEFT$(SP$,1)
5770 IFST$=""ORST$=";"THENPC=PC+1:BB$="IMP":GOTO5900
5780 R$=MID$(SP$,2,1):IFST$="A"AND(R$=" "ORR$=";"ORR$="")THENPC=PC+1:BB$="010":
TS=1:GOTO5900
5790 IFST$="#"ORST$="$"THENPC=PC+2:BB$="010":GOTO5900
5800 IFST$=","ANDLEFT$(SP$,2)=",X"THENPC=PC+3:BB$="111":GOTO5900
5810 IFST$=","ANDLEFT$(SP$,2)=",Y"THENPC=PC+3:BB$="110":GOTO5900
5820 IFST$="("THENPC=PC+3:BB$="011":GOTO5900
5830 IFST$="."ANDLEFT$(SP$,2)=".X"THENPC=PC+2:BB$="101":GOTO5900
5840 IFST$="."ANDLEFT$(SP$,2)=".Y"THENPC=PC+2:BB$="10":GOTO5900
5850 IFST$="."THENPC=PC+2:BB$="001":GOTO5900
5860 IFST$="/"ANDLEFT$(SP$,2)="/X"THENPC=PC+2:BB$="000":GOTO5900
5870 IFST$="/"ANDLEFT$(SP$,2)="/Y"THENPC=PC+2:BB$="100":GOTO5900
5880 IFST$="/"THENPC=PC+3:BB$="1":GOTO5900
5890 BB$="REL":PC=PC+2:GOTO5900
5900 :
5910 IFSA$="LDY"ANDBB$="10"THEN:K;BB$="101":GOTO5940
```

```
5920 IFSA$="LDX"ANDBB$="110"THENBB$="111":GOTO5940
5930 IFSA$="LDX"ANDBB$="10"THENBB$="110":FJ=1:GOTO5940
5940 READ C1$,C2$:IFC1$="-1"THEN5970
5950 IF C1$=SA$THEN5990
5960 GOTO5940
5970 :
5980 PRINT"]ERRORE IN";LA:END
5990 RESTORE
6000 M$=MID$(SA$,3,1):IFM$="X"ORM$="Y"THEN:IFBB$="010"THENBB$="000"
6010 VL$="":P=1:FORK=1TO8:IFMID$(C2$,K,1)="B"THENVL$=VL$+MID$(BB$,P,1):P=P+1:GO
TO6030
6020 VL$=VL$+MID$(C2$,K,1)
6030 NEXT:V=0:FORK=0TO7:IFVAL(MID$(VL$,K+1,1))=1THENV=V+2↑(7-K)
6040 NEXT:H=V:GOSUB6630
6050 IFBB$="IMP"OR(BB$="010"ANDTS=1)THENT=0:GOTO5570
6060 VM=1
6070 IFBB$="101"ORBB$="10"ORBB$="001"ORBB$="000"ORBB$="100"THENVM=0
6080 IFFJ=1ANDBB$="110"THENFJ=0:VM=0:REM 0,Y DI LDX
6090 IFBB$="REL"THEN6200
6100 M$=MID$(SA$,3,1):IFM$="X"ORM$="Y"THEN:IFBB$="000"THENBB$="010"
6110 IFBB$="010"THEN::GOTO6290
6120 GOSUB6420
6130 IF LEFT$(SP$,1)="#" OR LEFT$(SP$,1)="$"THEN6690
6140 FORN=1TOI:IFL$(N)=SP$THEN:GOTO6160
6150 NEXT:PRINT"]ERRORE IN";LA:END
6160 IFVL(N)>255ANDVM=0THENPRINT"]ERRORE IN";LA:END
6170 IFVL(N)=<255ANDVM=0THENH=VL(N):GOSUB6630:GOTO5570
6180 IFVL(N)=<255ANDVM=1THENH=VL(N):GOSUB6630:H=0:GOSUB6630:GOTO5570
6190 IFVL(N)>255THENH=VL(N)-256*INT(VL(N)/256):GOSUB6630:H=INT(VL(N)/256):GOSUB
6630:GOTO5570
6200 :
6210 FORK=1TOLEN(SP$):M$=MID$(SP$,K,1):IFM$=" "ORM$=";"THEN6230
6220 NEXT:GOTO6240
6230 SP$=MID$(SP$,1,K-1)
6240 FORK=1TOI:IFL$(K)=SP$THEN6260
6250 NEXT:PRINT"]ERRORE IN";LA:END
6260 D=ABS(PC-VL(K)):IFD>127THENPRINT"]ERR. RANGE DEL RELATIVO IN";LA:END
6270 IFPC>VL(K)THENH=256-(PC-VL(K)):GOSUB6630:GOTO5570
6280 H=VL(K)-PC:GOSUB6630:GOTO5570
6290 :
6300 X$=MID$(SP$,2,LEN(SP$)):IFASC(LEFT$(X$,1))<48ORASC(LEFT$(X$,1))>57THEN6330
6310 IF LEFT$(SP$,1)="$"THEN6340
6320 X=VAL(X$):H=X:GOSUB6630:GOTO5570
6330 :
6340 IFLEFT$(SP$,1)="$"THEN7270
6350 FORK=1TOLEN(X$):IFMID$(X$,K,1)=";"ORMID$(X$,K,1)=" "THEN6370
6360 NEXT:GOTO6380
6370 X$=MID$(X$,1,K-1)
6380 FORK=1TOI:IFL$(K)=X$THENH=VL(K):GOTO6400
6390 NEXT:PRINT"]ERRORE IN";LA:END
6400 IFH>255THENPRINT"]ERR. RANGE IN ";LA:END
6410 GOSUB6630:GOTO5570
6420 :
6430 FORK=1TOLEN(SP$):IFMID$(SP$,K,1)="("THEN6450
6440 NEXT:PRINT"]ERRORE IN";LA:END
6450 SP$=MID$(SP$,K+1,LEN(SP$))
6460 FORK=1TOLEN(SP$):IFMID$(SP$,K,1)=")"THEN6480
6470 NEXT:PRINT"]ERRORE IN";LA:END
6480 SP$=MID$(SP$,1,K-1)
```

```
6490 :RETURN
6500 :
6510 S$=MID$(S$,3,LEN(S$)):FORK=1TOLEN(S$):IFMID$(S$,K,1)="="THEN6530
6520 NEXT:PRINT"ERRORE IN";LA:END
6530 W1$=MID$(S$,1,K-1)
6540 W2$=MID$(S$,1+K,LEN(S$))
6550 IFLEFT$(W2$,1)="$"THEN7220
6560 IF LEFT$(W2$,1)="#"THEN6580
6570 PRINT"ERRORE IN";LA:END
6580 W2$=MID$(W2$,2,LEN(W2$)):FORK=1TOI
6590 IFW1$=L$(K)THENPRINT"ERRORE IN";LA:END
6600 NEXT
6610 I=I+1:VL(I)=VAL(W2$):L$(I)=W1$
6620 GOTO5120
6630 :
6640 :
6650 POKEJ+TP,H
6660 TP=TP+1:H$=STR$(H)
6670 PRINTH$,
6680 RETURN
6690 :
6700 IFLEFT$(SP$,1)="$"THENA$=MID$(SP$,2,4):GOSUB7110:H=V:M=1:GOTO6730
6710 M=ASC(MID$(SP$,2,3)):IFM<48ORM>57THENPRINT"ERRORE IN";LA:END
6720 H=VAL(MID$(SP$,2,LEN(SP$))):VM=1
6730 IFBB$="101"ORBB$="10"ORBB$="001"ORBB$="000"ORBB$="100"THENVM=0
6740 N=0:VL(0)=H:GOTO6160
6750 :
6760 S$=MID$(S$,3,LEN(S$))
6770 IFLEFT$(S$,1)="$"THEN7070
6780 IF LEFT$(S$,1)<>"#"THENPRINT"ERRORE IN";LA:END
6790 S$=MID$(S$,2,LEN(S$))
6800 PC=VAL(S$):J=PC:TP=0:IF FF=1THENPRINT:PRINT:RR=0:H$="===":GOSUB6670
6810 VOLTE=2:IF IN=1THEN IN=0:GOTO5570
6820 GOTO5120
6830 PRINT:PRINT"FINE ASSEMBLAGGIO.":FORK=0TO3500:NEXT:POKE198,0
6840 :
6850 PRINT"";CHR$(14);CHR$(8)
6851 PRINT"=================="
6852 PRINT"   ASSEMBLATORE   "
6853 PRINT"=================="
6860 PRINT:PRINT"1) ASSEMBLAGGIO":PRINT"2) SALVATAGGIO"
6870 PRINT"3) CARICAMENTO"
6890 PRINT"4) LISTARE SU VIDEO"
6902 PRINT"5) FINE PROGRAMMA ."
6903 PRINT" QUALE OPZIONE  ?";
6910 GETA$:IFVAL(A$)<1ORVAL(A$)>5THEN6910
6911 PRINT""
6930 IFA$="4"THENPRINT"":LIST-1110:END
6940 IFA$="1"THEN5001
6950 IFA$="5"THENPRINT"":END
6951 PRINT"STAI USANDO ":PRINT"IL PET (S/N)":INPUTOC$:IFLEFT$(OC$,1)="N"THEN69
60
6952 PRINT"LEGGERE L' ARTICOLO"
6953 FORYB=1TO6000:NEXT:RUN1111
6960 RESTORE:PRINT""
6970 READA$,B$:IFA$<>"-1"THEN6970
6980 INPUT"PERIFERICA (1/8)";P:INPUT"START (DEC)";V:INPUT"END (DEC)";W
6990 INPUT"NAMEFILE";V$:PRINT:PRINT:R=540:FORJ=1TO43:READT:POKER+J+5,T:NEXT:PO
KE549,P
```

```
 7000 IFP=8THENCLOSE1:IF W THEN OPEN1,8,15,"S0:"+V$
 7010 T=LEN(V$):POKE25,T:U=1:S=256*PEEK(50)+PEEK(49)+T
 7020 FORJ=1TOT:POKES-J,ASC(RIGHT$(V$,J)):NEXT:SYS546:U=V:T=252:GOSUB7060:IFW=0T
HEN7040
 7030 U=W:T=254:GOSUB7060:SYS579:RUN5000
 7040 FORJ=0TO5:POKER+J,PEEK(45+J):NEXT:SYS569:FORJ=0TO5:POKE45+J,PEEK(R+J):NEXT
 7050 RUN5000
 7060 POKET,INT(U/256):POKET-1,U-256*PEEK(T):RETURN
 7070 S$=MID$(S$,2,LEN(S$)):FORK=1TOLEN(S$):G$=MID$(S$,K,1):IFG$=" "ORG$=";"THEN
7090
 7080 NEXT
 7090 IFK<>3ANDK<>5THENPRINT"]ERRORE IN";LA:END
 7100 A$=MID$(S$,1,K-1):GOSUB7110:S$=STR$(V):GOTO6800
 7110 :
 7120 A$=MID$(A$,1,4):IFLEN(A$)<4THENGOSUB7150:V=A:RETURN
 7130 W$=A$:A$=MID$(W$,1,2):GOSUB7150:W=A
 7140 A$=MID$(W$,3,2):GOSUB7150:V=256*W+A:RETURN
 7150 A1$=LEFT$(A$,1):A2$=RIGHT$(A$,1)
 7160 IF A1$<"A" THEN7180
 7170 A1$=STR$(ASC(A1$)-55)
 7180 IF A2$<"A"THEN7200
 7190 A2$=STR$(ASC(A2$)-55)
 7200 A=VAL(A1$)*16+VAL(A2$)
 7210 RETURN
 7220 W2$=MID$(W2$,2,LEN(W2$)):FORK=1TOLEN(W2$):G$=MID$(W2$,K,1)
 7230 IFG$=" "ORG$=";"THEN7250
 7240 NEXT
 7250 IFK<>3ANDK<>5THENPRINT"]ERRORE IN";LA:END
 7260 A$=MID$(W2$,1,K-1):GOSUB7110:W2$="#"+STR$(V):GOTO6580
 7270 FORK=1TOLEN(X$):G$=MID$(X$,K,1)
 7280 IFG$=" "ORG$=";"THEN7300
 7290 NEXT
 7300 IFK<>3ANDK<>5THENPRINT"]ERRORE IN";LA:END
 7310 A$=MID$(X$,1,K-1):GOSUB7110:H=V:GOTO 6400
 7320 DATAADC,011BBB01,AND,001BBB01,ASL,000BBB10,BCC,10010000,BCS,10110000,BEQ,1
1110000
 7330 DATABIT,001BBB00,BMI,00110000,BNE,11010000,BPL,00010000,BRK,00000000,BVC,0
1010000
 7340 DATABVS,01110000,CLC,00011000,CLD,11011000,CLI,01011000,CLV,10111000,CMP,1
10BBB01
 7350 DATACPX,111BBB00,CPY,110BBB00,DEC,110BB110,DEX,11001010,DEY,10001000,EOR,0
10BBB01
 7360 DATAINC,111BB110,INX,11101000,JMP,01B01100,JSR,00100000,LDA,101BBB01
 7370 DATALDX,101BBB10,LDY,101BBB00,LSR,01BBB110,NOP,11101010,ORA,000BBB01,PHA,0
1001000
 7380 DATAPHP,00001000,PLA,01101000,PLP,00101000,ROL,001BBB10,ROR,011BBB10,RTI,0
1000000
 7390 DATARTS,01100000,SBC,111BBB01,SEC,00111000,SED,11111000,SEI,01111000,STA.1
00BBB01
 7400 DATASTX,100BB110,STY,100BB100,TAX,10101010,TAY,10101000,TSX,10111010,TXA,1
0001010
 7410 DATATXS,10011010,TYA,10011000,INY,110010000,-1,-1
 7420 DATA169,1,162,1,160,0,32,186,255,165,25,166,49,164,50,32,189,255,169,000
 7421 DATA 133,157
 7430 DATA96,169,0,166,251,164.252,32,213,255,96,169,251,166,253,164,254,32,216,
255,96
 7440 REM FORZA KEY
 7450 PRINT#3:CLOSE3:   RUN1111
READY.
```

Programmi di prova per il Vic 20

Un bip ad ogni return

```
10 "*$033C;
11 ";
12 ";INIZIALIZZAZIONE
13 ";
14 "    LDA (#772);
15 "    STA (#1000);
16 "    LDA (#773);
17 "    STA (#1001);
18 "    LDA #132;
19 "    STA (#772);
29 "    LDA #3;
30 "    STA (#773);
31 "    RTS ;
32 ";
33 ";GENERA SONORO
34 ";
35 "*#900;
36 "=VOL=#36878;
37 "=SON=#36875;
38 "=NOTA=#211;
39 "    STA .(#1);
40 "    STX .(#2);
41 "    STY .(#3);
42 "    LDA #15;
43 "    STA (VOL);
44 "    LDA #NOTA;
45 "    STA (SON);
46 "    LDY #255
47 "@UNO;
48 "    LDX #255
49 "@DUE;
50 "    DEX ;
51 "    BNE DUE;
52 "    DEY ;
53 "    BNE UNO;
54 "    LDA #0;
55 ";   STA (VOL);
56 "    LDA .(#1);
57 "    LDX .(#2);
58 "    LDY .(#3);
59 "    JMP /(#1000);
60 ";
61 ";
```

Per lo scroll del video

```
10 "*#828
11 ";
12 ";
14 "    LDX #230
15 "@UNO
16 "    LDA ,X(#4349)
17 "    STA ,X(#4371)
18 "    LDA ,X(#38141)
19 "    STA ,X(#38163)
20 "    DEX
21 "    CPX #255
22 "    BNE UNO
23 ";
24 "    LDX #252
25 "@DUE
26 "    LDA ,X(#4096)
27 "    STA ,X(#4118)
28 "    LDA ,X(#37888)
29 "    STA ,X(#37910)
30 "    DEX
31 "    CPX #255
32 "    BNE DUE
33 ";
34 "    LDX #22
35 "    LDA #32
36 "@TRE
37 "    STA ,X(#4095)
38 "    DEX
39 "    BNE TRE
40 "    RTS
41 ";
42 ";
```

Per rinumerare

```
100 "*#828
102 "    LDA #10
104 "    STA .(#4)
105 "    LDA #0
106 "    STA .(#5)
107 "    LDA #1
108 "    STA .(#0)
109 "    LDA #18
110 "    STA .(#1)
111 "@DUE
112 "    LDY #0
113 "    LDA /Y(#0)
114 "    BNE UNO
115 "    RTS
116 "@UNO
117 "    STA .(#6)
118 "    INY
119 "    LDA /Y(#0)
120 "    STA .(#7)
121 "    LDA .(#4)
122 "    INY
123 "    STA /Y(#0)
124 "    LDA .(#5)
125 "    INY
126 "    STA /Y(#0)
127 "    CLC
128 "    LDA .(#4)
129 "    ADC #10
130 "    STA .(#4)
131 "    BCC TRE
132 "    LDA .(#5)
133 "    ADC #0
134 "    STA .(#5)
135 "@TRE
136 "    LDA .(#6)
137 "    STA .(#0)
138 "    LDA .(#7)
139 "    STA .(#1)
140 "    LDA #0
141 "    BEQ DUE
142 ";
```

Organetto

```
10 "*$033C
11 ";
12 "=ON=$0F
13 "=OFF=$00
14 "=KEY=$CB
15 "=F7=$BF
16 "=NT=$40
17 ";
18 "    LDA #ON
19 "    STA ($900E)
20 "@UNO
21 "    LDA .(KEY)
22 "    CMP #NT
23 "    BEQ UNO
24 "    ADC $80
25 "    STA ($900B)
26 "    CMP #F7
27 "    BNE UNO
28 "    LDA #OFF
29 "    STA .($C6)
30 "    STA ($900E)
31 "    RTS
```

COMMOD

commodore MMF 9000 computer

CBM

Se stai comprando un personal computer prova a farti queste domande:
1. Chi è oggi il più affidabile?
2. Chi dà la possibilità di scegliere fra più sistemi?
3. Chi fornisce soluzioni, subito, in una gamma vastissima?
4. Chi propone il miglior rapporto fra costi e prestazioni?
5. Chi ti dà una così grande esperienza ed assistenza?
A tutte le domande puoi rispondere con

ORE, IL N°1

un solo nome: Commodore Computer.
Anche per questo Commodore
è il Numero 1. In Europa e in Italia.
Sei in buone mani.
Commodore Italiana Spa
Milano, telefono 02/6125651

10 "@routine

Il precedente esempio definisce l'etichetta "routine" utile per esempio, nel caso si voglia scrivere un relativo diretto ad un indirizzo variabile:

es: 10 "bne routine

Il numero massimo di label disponibili è scritto nel doppio dimensionamento di vettori di riga 5010. Nel caso si vogliono più di 8 etichette, per esempio, in numero x, si dovrà digitare:

5010 clr:dim l$(x), vl(x): print " [illegible] "

Si ricordi, inoltre, che nello scrivere una linea, tra l'istruzione mnemonica (es: lda) e il relativo argomento (es: #234), occorre inserire uno spazio.

La capacità di riferirsi a valori decimali (#) oppure esa ($), si aggiunge alla possibilità di definire delle costanti, di più lettere e in qualsiasi punto del programma. Per esempio, volendo assegnare a k il valore decimale 1234, si scriverà:

10 "=k=#1234, eventuale commento.

Per utilizzare tale costante, per esempio, in modo assoluto, si deve scrivere:

10 " lda (k) di cento più vantaggioso di:

10 " lda (#1234)

Chiarito come va scritto il programma da assemblare, vediamo come gira il vero e proprio assemblatore.

Come detto precedentemente, occorre digitare "run 1111"; in seguito tale ordine il video mostra una serie di opzioni, ottenibili con la pressione dei tasti-funzione, 1,3,5,7, ed 8, come appunto, specificato nel menu. Vediamole singolarmente.

Assemblaggio

Premendo (f1) l'assemblatore effettua una prima verifica, riga per riga, in modo da controllare che tutte le labels siano corrette e che siano, altresì, esatti gli assegnamenti delle costanti. Il video mostra due colonne di valori: la prima indica il numero della linea che sta per essere esaminata, la seconda è la successione delle locazioni di memoria utilizzate dall'assemblatore per allocare il programma; in seguito lo schermo mostra tutte le labels e costanti, con il relativo valore espresso in decimale.

Premendo la barra spaziatrice, l'assemblatore provvede ad un ulteriore test e solo se tutto è corretto comincia a tradurre l'assembler in Linguaggio Macchina (L.M.).

Nel caso si riscontrino errori, verrà segnalato il numero della riga non corretta, col conseguente arresto della fase di traduzione.

In tal caso, bisogna verificare che la sintassi di tale riga sia esatta, se in essa siano state usate costanti non nominate precedentemente, se il salto del relativo sia troppo esteso, se il relativo chiami una label innominata, e così via,

Eseguita la correzione necessaria, riprendere nuovamente la traduzione, mediante
"run 1111"
Corretti eventuali errori, terminato l'assemblaggio vero e proprio, il programma assemblatore mostra sullo schermo la primitiva videata, che avverte delle diverse opzioni.

I tasti (f3) o (f5), provvedono alla fase di lettura/scrittura su nastro/disco del programma precedentemente assemblato.

Facciamo l'ipotesi che la nostra routine in l.m. sia allocata da 828 (dec) sino a 1000 (dec). Volendo registrare su nastro o disco tale zona, scelta la periferica (nastro=1 / disco=8) bisognerà rispondere agli input nel modo seguente:

start? 828
end? 1000

Per ricaricare invece da periferica il programma in l.m., precedentemente registrato, si risponderà:

start? 828
end ? 0

Listati

Digitando (f7) si otterà sul video l'intero listato del programma da assemblare.

Desiderando invece il list su stampante, si premerà (f8).

Uscita

Alla pressione del tasto-funzione n.5, si esce dal programma assemblatore, cosa utile (e più elegante che premere restore + run/stop), nel caso si vogliano apportare modifiche o aggiunte al list del programma da assemblare. ■

IMPARA A PROGRAMMARE CON IL VIC

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

★ DISPENSA N. 7 ★

Scrivi anche tu

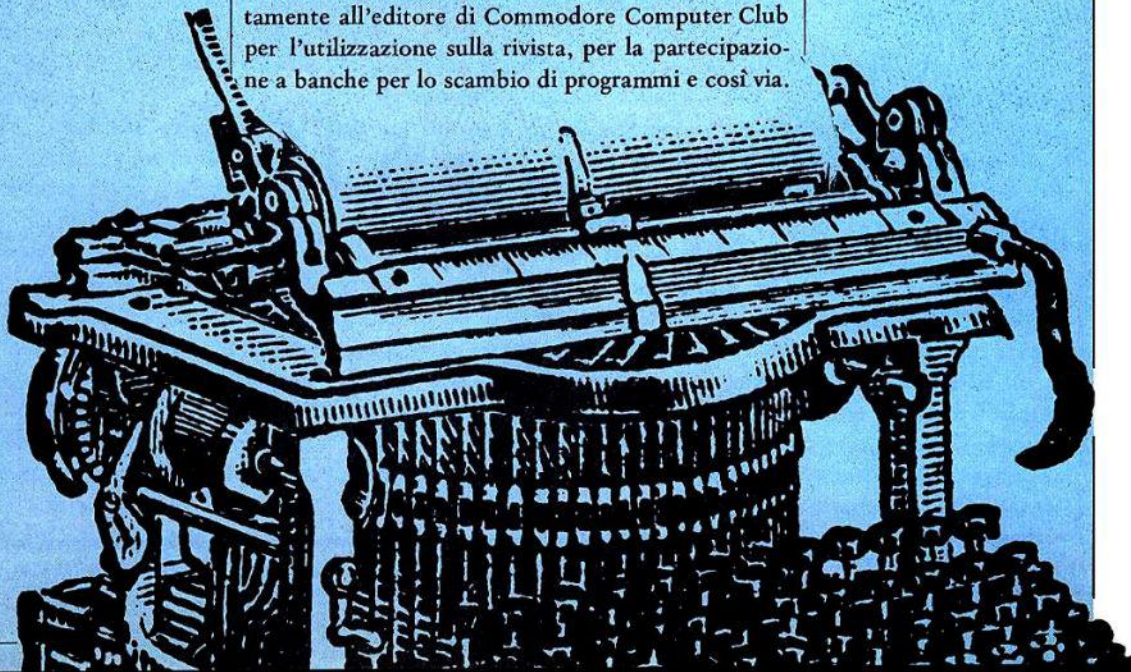
La collaborazione a Commodore Computer Club è aperta a tutti gli utenti di sistemi CBM che abbiano sviluppato programmi e routine originali e d'interesse generale o che desiderino fare partecipi gli altri lettori delle scoperte e delle osservazioni fatte per una migliore utilizzazione di queste macchine.

Allegate un breve articolo in cui spiegate a che serve il programma, come funziona, su quale computer gira e con quale configurazione (espansa, non espansa, ecc.).

- Iniziate tutti i listati con delle Rem indicanti il titolo del programma, il nome, l'indirizzo ed il telefono dell'autore.
- Inviate i listati su disco o su cassetta, (su carta verranno irrimediabilmente cestinati).

Per compensarvi per questa spesa la redazione vi spedirà a stretto giro di posta due cassette e/o dischetti nuovi per ciascuna cassetta da voi inviata, indipendentemente dal fatto che il programma registrato venga accettato per la pubblicazione.

Ciascun articolo e/o programma pubblicato, invece, verrà compensato con l'invio d'una intera scatola di cassette e/o dischetti.

Gli autori, naturalmente, si assumono ogni responsabilità circa l'originalità dei lavori inviati i cui diritti di riproduzione si intendono ceduti illimitatamente all'editore di Commodore Computer Club per l'utilizzazione sulla rivista, per la partecipazione a banche per lo scambio di programmi e così via.

Il suono

Il sistema acustico VIC

Il sistema acustico del VIC è costituito da quattro operatori acustici indipendenti con un intervallo di frequenza di tre ottave ciascuno. Tre dei quattro generatori danno un suono "puro", mentre il quarto genera un "disturbo bianco", termine con il quale si indica il disturbo che ha un ampio spettro di frequenza e viene usato per creare una vasta gamma di effetti acustici, come ad esempio, le esplosioni, che sarebbero difficili da sintetizzare partendo dalla serie dei tre suoni puri disponibili.

Inoltre, il sistema permette il controllo del volume interamente da programma e poichè il suono non può essere provato se non con il volume alzato, cominciamo proprio da qui.

Controllo del volume

Il livello del suono è stabilito fissando con l'istruzione POKE a 36878 un valore compreso fra 0 ae 15. Lo zero rappresenta il volume più basso, il 15 il più alto. Il livello acustico relativo è fissato dalla manopola del volume che si trova sul vostro televisore. Notate che il VIC agisce sui valori interi fornendo quindi 16 livelli acustici (comprendendo lo zero), cosicchè accendendo con il comando POKE alla locazioe di memoria 36878 e inserendo il valore 4.99, si imposta il volume al livello 4.

Controllo delle tonalità

I tre generatori di suoni puri vengono attivati accedendo con il comando POKE alle locazioni 36874, 36875 e 36876, il generatore di disturbi, accedendo alla locazione 36877. I valori che possono essere inseriti con il comando POKE sono compresi fra 128 e 255 inclusi. I valori fuori limite o non producono alcun effetto o generano un errore di sintassi. Per spegnere un certo generatore di suoni, è comunque consigliabile utilizzare il valore 0. Notate che l'inserimento dello stesso valore nelle locazioni 36874, 36875 e 36876 provoca tre suoni diversi e che i suoni si alzano con i valori da 128 e 254 mentre il valore 255 provoca la generazione di un tono basso.

```
10 REM SUONO!!!
20 GOSUB 1000
30 GET A$: IF A$="" THEN 30
40 IF A$=" " THEN END
50 GOTO 20
60 :
70 :
80 :
1000 REM DUE NOTE
1010 POKE 36878,15
1020 FOR N=1 TO 5
1030 POKE 36874,240
1040 FOR S=1 TO 375
1050 NEXT
1060 POKE 36874,230
1070 FOR S=1 TO 375
1080 NEXT S
1090 NEXT N
1100 POKE 36874,0
1110 POKE 36878,0
1120 RETURN
1130 :
1140 :
```

Tabella 1 Routine a due tonalità

Nelle tabella 1 è riportato un semplice pro-

gramma per l'alternanza di due note. La linea 1010 attiva il tutto volume, le linee 1020 e 1090 provocano le ripetizione per dieci volte dei due toni, la linea 1030 attiva il valore di nota 240 e il loop FOR (dalla linea 1040 alla linea 1050) produce un ritardo che indica per quanto tempo deve essere mantenuta la nota. La nota alternativa (valore 230) viene quindi suonata e mantenuta con le stesse modalità dalle linee 1060, 1080 e alla fine dei dieci cicli, la linea 1100 spegne la nota e la linea 1110 abbassa completamente il volume.

E' importante concludere qualsiasi routine per la generazione di suoni con istruzioni che interrompono il segnale acustico e riportano il volume a zero. Non facendolo, per il fatto di utilizzare il comando POKE, le locazioni cui si è acceduto conservano il loro stato che influenza eventuali programmi successivi. Questa considerazione è altrettanto importante nell'ambito dello stesso programma in quanto gli effetti acustici vengono per lo più incorporati sotto forma di subroutine. Tutti i programmi di questo capitolo sono stati scritti sotto questa forma e per questa ragione hanno come ultima linea un'istruzione RETURN.

Controllo del tono e del volume

Nel paragrafo abbiamo usato una versione più completa della routine precedente che simula il passaggio di una sirena. Il programma nella tabella 10.2 dimostra il modo in cui può essere variato il livello acustico e può essere modificata la nota per mezzo di variabili di manipolazione.

Il loop FOR, che va dalla linea 2070 alla linea 2140, serve a controllare il livello acustico. Stabilendo per le variabili X, Y e Z rispettivamente i valori 1, 15 e 1, prima del loop, si ottiene un incremento del livello in corrispondenza della linea 2080 (POKE 36878,L) per ogni passaggio del loop a partire da 1 fino a 15, con il risultato di ottenere che il suono dentro il loop diventi sempre più alto. Non appena viene raggiunto il volume massimo (dalla linea 2070 alla linea 2140), il loop viene abbandonato mentre vengono modificati i suoi parametri in corrispondenza delle linee 2150, 2160 e 2170 per fare in modo che, la prossima volta in cui verrà eseguito, decrementi L da 15 a 0. Questo secondo passaggio del loop è ottenuto per mezzo del loop J (dalla linea 2060 alla linea 2180).

```
2000 REM SIRENA CHE PASSA
2005 REM E SI ALLONTANA
2010 T=240
2020 D=10
2030 X=1
2040 Y=15
2050 Z=1
2060 FOR J=1 TO 2
2070 FOR L=X TO Y STEP Z
2080 POKE 36878,L
2090 POKE 36875,T
2100 FOR S=1 TO 375
2110 NEXT S
2120 D = -D
2130 T = T+D
2140 NEXT L
2150 X=15
2160 Y=0
2170 Z = -1
2180 NEXT J
2190 POKE 36875,0
2200 RETURN
```

Tabella 2 Passaggio di una sirena

Nel programma precedente (tabella 1), i due suoni sono stati prodotti da due diverse istruzioni POKE e due serie di loop ritardanti identici. Per i programmi più lunghi, esistono metodi di programmazione alternativi che consentono di evitare le ripetizione di linee di codice simili fra loro. Il programma che stiamo esaminando (tabella 10.2) illustra come funziona uno dei metodi alternativi possibili per la produzione di una coppia di suoni.

Uno dei due valori di tonalità richiesti dal problema viene dichiarato come T nella linea 2010, mentre nella linea 2020 viene dichiarata come D la differenza fra i due valori previsti. Nel loop di controllo del volume principale il suono è generato con la linea 2090 e mantenuto per tutta la durata del loop di sostegno (dalla linea 2100 alla linea 2110). Le linee della 2120 alla 2130 provocano la modifica del valore di tono prima del passaggio successivo. Più in particolare, la prima fa sì che la differenza di tonalità sia alternativamente +D, -D, +D, -D e

QUANTI COLORI HA LA TUA STAMPANTE ?

NEL 1983 LA SEIKOSHA PER PRIMA AL MONDO
E' IN GRADO DI PRESENTARE LA NUOVA STAMPANTE
GRAFICA A SETTE COLORI.

RIUNITE IN UN APPARECCHIO PRATICO E COMPATTO
LE CARATTERISTICHE DELLA STAMPANTE E DEL PLOTTER,
LA SEIKOSHA INVENTA UN NUOVO TIPO DI PERIFERICA
CHE BEN PRESTO SARA' INSOSTITUIBILE.

REBIT COMPUTER E' ORGOGLIOSA DI LANCIARE
QUESTA NOVITA' ASSOLUTA SUL MERCATO ITALIANO
AD UN PREZZO MOLTO, MOLTO COMPETITIVO:
MENO DI UN MILIONE.
MENO DI UNA COMUNE STAMPANTE IN BIANCONERO.

GP-700A

Graphic Color Printer

SEIKOSHA

Comandi per Disco — OPEN X, 8, 15			
Load "Nome", 8	Carica il prog. Basic: "Nòme"	"IØ"	Inizializza il disco
Load "Nome", 8,1	Carica il prog. in Ling. Macc. "Nome"	"VØ"	Cancella i file impropri
Save "Nome", 8	Registra su disco il prog: "Nome"	"CØ: unico = Ø: parte 1,0: parte 2"	Concatena 2 file in un sc
Save "@Ø: Nome"	Registra, sovrapp., il prog: "Nome"	"CØ: nuovo = Ø: vecchio"	Duplica il "Vecchio" file
Verify " * ",8	Verifica l'ultimo programma	"RØ: nuovo = vecchio"	Cambia in "Nuovo" il n
"NØ: intestazione, XX"	Formatta il disco con ID = XX	"SØ: primo, secondo"	Cancella i file: "Primo"

LEGENDA	
Comando	Abbrev.
Codice Commodore (decim.)	Codice Commodore (esadec.)

ABS aB 182 B6	AND aN 175 AF	ASC aS 198 C6	ATN aT 193 C1	CHR$ cH 199 C7	CLOSE clO 160 A0	CLR cL 156 9C	CMD cM 157 9D	CON 154
DEF dE 150 96	DIM dI 134 86	END eN 128 80	EXP eX 189 BD	FOR fO 129 81	FRE fR 184 B8	GET gE 161 A1	GOSub goS 141 8D	GOT 137
LET lE 136 88	LEFT$ leF 200 C8	LIST lI 155 9B	LOAD lO 147 93	MID$ mI 202 CA	NEXT nE 130 82	NOT nO 168 A8	OPEN oP 159 9F	PEE 194
PRINT ? 153 99	PRINT# pR 152 98	READ rE 135 87	Restore reS 140 8C	Return reT 142 8E	RIGHT$ rI 201 C9	RND rN 187 BB	RUN rU 138 8A	SAV 148
SIN sI 191 BF	SPC(sP 166 A6	SQR sQ 186 BA	STEP stE 169 A9	STOP sT 144 90	STR$ stR 196 C4	SYS sY 158 9E	TAB(tA 163 A3	THE 167
VAL vA 197 C5	VERIFY vE 149 95	WAIT wA 146 92	COMANDI SENZA ABBREVIAZIONE			COS 190 BE	FN 165 A5	GET 161+
INPUT 133 85	INT 181 B5	LEN 195 C3	LOG 188 BC	NEW 162 A2	ON 145 91	OR 176 B0	POS 185 B9	REM 143

Reset del sistema: SYS(x)

VIC 20.X = 64802

CBM64.X = 64738

LEGENDA									
COLORE / CHR$	CARAT. SPEC.	C=	Aranc. 192	Marrone 149	Rosa 150	Grigio 1 151	Grigio 2 152	Verdino 153	Celeste 154	Grigio 3 155
SCHERMO	SOLO PER VIC 20		8	9 Paglierino	10	11 Celeste	12 Viola	13	14	15 Giallino
		CTRL	Nero 144	Bianco 5	Rosso 28	Azzurro 159	Viola 156	Verde 30	Blu 31	Giallo 158	Rev 18
			0	1	2	9	4	5	6	7	

VEDI CHR$ (8) CHR$ (9)

CHR$ (14) CHR$ (142)

PUNTATORI PIU' USATI		
LO	HI	RELATIVI A:
43	44	START BASIC
45	46	START VARIABILI
47	48	START ARRAY
49	50	FINE ARRAY
51	52	STRINGHE
53	54	FINE STRINGHE
55	56	TOP OF MEMORY

SUONO COMMODORE 64

54272 54273	NOTE MUSICALI	Ø ÷ 255
54276 FORME D'ONDA	17 = TRIANG. 33 = SINUSOI. 64 = PULSAN. 129 = RUMORE	
54274 SOLO PER 54275 ONDE PULS.		Ø ÷ 255 / Ø ÷ 15

LOCAZIONI UTILI

	BIT 4 ÷ 7	BIT Ø ÷ 3	
54277	ATTACK	DECAY	Ø ÷ 255
54278	SUSTAIN	RELEASE	Ø ÷ 255

54296 VOLUME Ø ÷ 15

N.B. X = PRIMA VOCE

X + 7 = SECONDA VOCE

X + 14 = TERZA VOCE

SUONO

X = 36874.Y

X = 36875.Y

X = 36876.Y

X = 36877.Y

X = 36878.Y

ropri
un solo "Unico"
"filecol nome"Nuovo"
" il nome del"Vecchio" file
mo" "Secondo"

CONT	cO		DATA	dA
154	9A		131	83
GOTO	gO		INPUT#	iN
137	89		132	84
PEEK	pE		POKE	pO
194	C2		151	97
SAVE	sA		SGN	sG
148	94		180	B4
THEN	tH		USR	uS
167	A7		183	B7
GET#			IF	
161+35	A1+23		139	8B
REM			TAN	
143	8F		192	C0

Solo CBM 64

CBM 64 e VIC 20

Rev. 18 R — Rev. Off. 146

PRINT CHR$ (X)

5 Bianco	37 %	55 7	73 I	91 [	109	127
8 Ca. Blocco	38 &	56 8	74 J	92 £	110	133
9 Ca. Sblocco	39 '	57 9	75 K	93]	111	134
13 Return	40 (	58 :	76 L	94 ↑	112	135
14 Min-Mai	41)	59 ;	77 M	95 ←	113	136
17 CRSR↓	42 *	60 <	78 N	96	114	137
18 RVS.ON	43 +	61 =	79 O	97	115	138
19 Home	44 ,	62 >	80 P	98	116	139
20 Del	45 –	63 ?	81 Q	99	117	140
28 Rosso	46 .	64 @	82 R	100	118	141 Shift+ Return
29 CRSR→	47 /	65 A	83 S	101	119	142 Maius.
30 Verde	48 0	66 B	84 T	102	120	144 Nero
31 Blu	49 1	67 C	85 U	103	121	145 Curs.↑
32 Spazio	50 2	68 D	86 V	104	122	146 RvsOff
33 !	51 3	69 E	87 W	105	123	147 Home
34 "	52 4	70 F	88 X	106	124	148 Ins.
35 #	53 5	71 G	89 Y	107	125	156 Viola
36 $	54 6	72 H	90 Z	108	126 π	157 ← CRSR

158 Giallo	176	184	129 Aranc.
159 Azzur.	177	185	149 Marr.
160 Spazio	178	186	150 Rosa
161	179	187	151 Grigio 1
162	180	188	152 Grigio 2
163	181	189	153 Verdino
164	182	190	154 Celeste
165	183	191	155 Grigio 3
166			
167			
168			
169			
170			
171			
172			
173			
174			
175			

192 ÷ 223 Vedi 96 ÷ 127
224 ÷ 254 Vedi 160 ÷ 190
255 = 126

LEGENDA: CHR$
- Codice Ascii
- Caratteri Grafici
- Colori aggiuntivi CBM 64
- Comandi speciali

NORM + SHIFT

Cancella schermo 147	Inserim. caratt. 148
Cursore in alto 19	Cancella caratt. 20

Norm + Shift

F 1 133	F 2 137
F 3 134	F 4 138
F 5 135	F 6 139
F 7 136	F 8 140

A. DE SIMONE

NORM + SHIFT

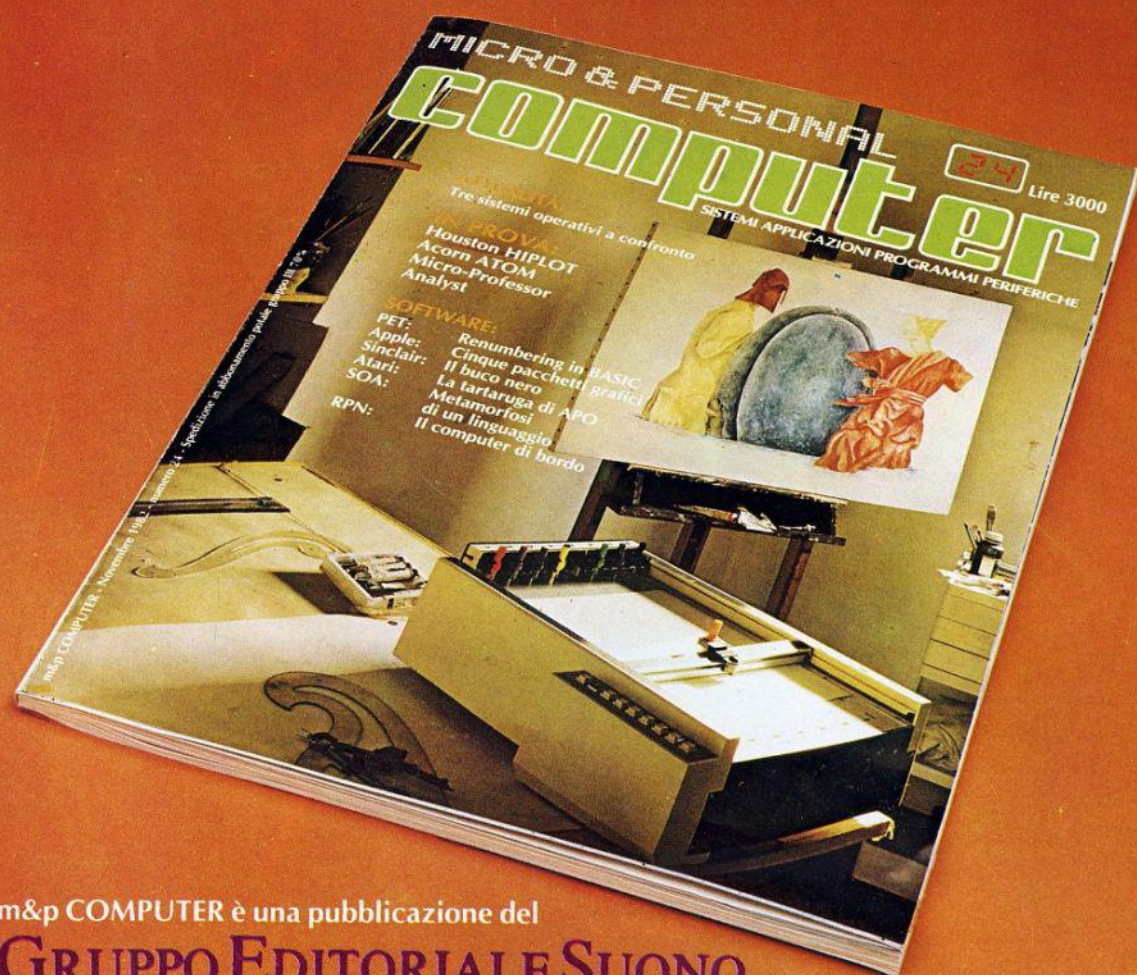
CURSOR ↑ 145	CURSOR ← 157
CURSOR ↓ 17	CURSOR → 29

NO VIC 20 (POKE X, Y)

4.Y = 128 ÷ 255 Prima tonalità
5.Y = 128 ÷ 255 Seconda tonalità
6.Y = 128 ÷ 255 Terza tonalità
7.Y = 128 ÷ 255 Rumore bianco
8.Y = Ø ÷ 15 Volume

COLORI	BORDO	FONDO	Val. Standard
VIC 20	36879 BIT Ø ÷ 3	36879 BIT 4 ÷ 7	27
CBM 64	53280	53281	254 246

VIC 20		Versione base	+ 3K RAM	+8 ÷ 16 K
Basic	#	4096-7679	1024-7679	4608 ÷
	$	1000 1DFF	0400-1DFF	1200 ÷
Schermo	#	7680-8191	7680-8191	4096-4607
	$	1E00-1FFF	1E00-1FFF	1000-11FF
Colore	#	38400-38911	38400-38911	37888-38399
	$	9600-97FF	9600-97FF	9400-95FF
CBM 64		BASIC	SCHERMO	COLORE
	#	2048-40959	1024-2023	55296-56295
	$	0800-9FFF	0400-07E7	D800

ENTRA NELLA NUOVA DIMENSIONE... LEGGI

m&p COMPUTER è una pubblicazione del

GRUPPO EDITORIALE SUONO

Via del Casaletto 380 - 00151 Roma

così via ad ogni passaggio successivo, mentre la linea 2130 modifica il valore di tonalità alternativamente nei due valori richiesti. Notate che al primo passaggio, la variabile T viene modificata "in giù", quindi il valore iniziale di T (linea 2010) ha bisogno di essere la più alta fra le due note.

La penultima linea (2190) spegne il generatore di suoni, mentre il volume è già completamente dato che il valore finale di L nel loop esistente è uguale a zero.

Effetti acustici

La generazione di effetti acustici dipende dal controllo software dei relativi parametri. Benchè siano possibili infinite variazioni, le caratteristiche principali da tenere in considerazione sono le seguenti:

a) le frequenze e la loro progressione nel tempo;
b) la combinazione delle frequenze;
c) il volume e le sue variazioni nel tempo;
d) il rapporto suono-silenzio.

La frequenza può essere progressivamente modificata mediante il loop FOR. La progressione in basso delle frequenze, come ad esempio quella prodotta da un oggetto in caduta, può essere ottenuta a partire dal programma riportato nella tabella 3.

```
3000 REM OGGETTO CHE... CADE
3010 POKE 36878,15
3020 FOR T=240 TO 130 STEP -1
3030 POKE 36876,T
3040 FOR S=1 TO 30
3050 NEXT S
3060 NEXT T
3070 POKE 36876,0
3080 POKE 36878,0
3090 RETURN
```

Tabella 3 Un oggetto in caduta

Una combinazione di frequenze si può ottenere per esempio modificando il loop FOR alla linea 3020 in modo che si presenti nella forma:

3020 FOR T = 130 TO 240

simulando la scomparsa di un oggetto con un improvviso aumento delle frequenze.

Le variazioni di volume sono già state viste nel programma di simulazione del passaggio di una sirena (tabella 10.2). Un effetto sonoro spesso utile è quello dell'esplosione che può essere prodotta utilizzando il cosiddetto "disturbo bianco" (cioè POKE 36877) e facendolo morire lontano con un programma come quello riportato nella tabella 4.

```
4000 REM ESPLOSIONE
4010 POKE 36877,140
4020 FOR L=15 TO 0 STEP -1
4030 POKE 36878,L
4040 FOR S=1 TO 150
4050 NEXT S
4060 NEXT L
4070 POKE 36877,0
4080 RETURN
4090 :
```

Tabella 4 L'esplosione

Il rapporto suono-silenzio è importante per effetti sonori come ad esempio quello prodotto dal telefono e simulato con il programma riportato nella tabella 5.

La prima linea (5010) produce un accensione a tutto volume che viene abbassato completamente con l'ultima linea (5160) prima del RETURN.

```
5000 REM SQUILLO DI TELEFONO
5005 :
5010 POKE 36878,15
5020 FOR R=1 TO 5
5030 FOR N=1 TO 2
5040 FOR T=1 TO 20
5050 POKE 36876,238
5060 FOR S=1 TO 5
5070 NEXT S
5080 POKE 36876,0
5090 NEXT T
5100 FOR D=1 TO 30
5110 NEXT D
5120 NEXT N
5130 FOR P=1 TO 1000
5140 NEXT P
5150 NEXT R
5160 POKE 36878,0
5170 RETURN
5180 :
```

Tabella 5 Il telefono

Il loop FOR esterno (dalla linea 5020 alla linea 5150) ripete il suono di base per cinque volte, le linee dalla numero 5030 alla numero 5120 forma-

no un altro loop FOR, che produce due trilli in rapida successione e, infine, le linee 5130 e 5140 formano una pausa relativamente lunga prima che l'intero ciclo venga ripetuto.

Nel loop FOR il suono puro viene modificato (dalla linea 5040 alla linea 5090) in un effetto simile al trillo, mediante la generazione di 20 battute dello stesso suono, sostenuta ciascuna nelle linee dalla 5060 alla 5070 per un breve periodo di tempo.

In questo capitolo, abbiamo visto solo alcuni degli effetti acustici che è possibile ottenere con il VIC e, in tutti gli esempi riportati, il cambiamento progressivo dei modi di controllo da parte del loop FOR del volume o della frequenza era lineare. Esistono invece altri interessanti effetti che possono essere prodotti con forme di progressivi esponenziali o di altro tipo o variando con la funzione RND, volume, frequenza e/o tempi.

Esercizi

Scrivete un programma che produca le note seguenti:

(I numeri fra parentesi indicano la durata relativa di ciascuna nota). Provate inoltre a cambiare i colori sulla base delle note suonate. I valori del comando POKE per le note musicali sono riportati nell'Appendice G. Il programma che permette di raggiungere lo scopo è riportato nell'Appendice A18.

Tastiera

Un programma musicale che potrebbe interessare gli utenti del Vic

LE capacità musicali del Vic possono essere sfruttate in moltissimi programmi; in generale, però, la difficoltà principale che si incontra consiste nell'associare i tasti del computer alle note da suonare.

Ecco un programma che visualizza sullo schermo una tastiera con 15 note (dal do al mi dell'ottava successiva) con possibilità d'uso delle trè tonalità, mediante i tasti F1, F3, F5.

Premendo i tasti relativi alle note si vedrà contemporaneamente una pallina spostarsi, indicando la nota che viene suonata in quel momento.

Con un po' di pratica sarà quindi possibile suonare guardando solo la tastiera sullo schermo.

Spiegazioni sul listato:

• le linee 10-60 unitamente al sottoprogramma 2000 e al blocco dati 3000-3004 fanno apparire sullo schermo la tastiera;

• le linee 74-108 richiamano i sottoprogrammi 1000-1170, che generano i suoni e fanno spostare la pallina;

• le linee 110-114 richiamano i sottoprogrammi 1180-1200, che cambiano la tonalità.

Se le note non dovessero corrispondere ad altri strumenti da voi in possesso, agire sulle linee 1000, 1010, 1020,..., 1170. ▪

Giampaolo Cervone
Via Vicenzo Riolo, 10 - 90139 Palermo
Tel. 580061

```
1 REM ***        TASTIERA MUSICALE      ***
2 REM *** PER VIC 20 SENZA ESPANSIONI ***
3 REM *** GIANPAOLO CERVONE -PALERMO- ***
4 REM *** TEL. 580061 VIA V.RIOLO N.10***
5 POKE36879,93:PRINT"[illegible]"
10 CA=7680:CO=38400
11 V=36878:S=36876
12 J=99
15 FORK=286TO306
20 GOSUB2000:NEXT
22 J=93
25 FORK=196TO284STEP2
30 GOSUB2000:NEXT
35 READK:IFK=-1GOTO47
40 GOSUB2000
45 GOTO35
47 J=160
50 READK:IFK=-1GOTO67
55 GOSUB2000
60 GOTO50
67 PRINT"[illegible] W E  T Y U  O P"
68 PRINT" A S D F G H J K L :"
69 POKEV,15
70 PRINT"[illegible]F1=ALTI":PRINT
71 PRINT"F3=MEDI":PRINT
72 PRINT"F5=BASSI":PRINT
```

➔

```
73 PRINT"SPAZIO=STOP"
74 GETA$
75 IFA$="A"THENGOSUB1000
78 IFA$="W"THENGOSUB1010
80 IFA$="S"THENGOSUB1020
82 IFA$="E"THENGOSUB1030
83 IFA$="D"THENGOSUB1040
84 IFA$="F"THENGOSUB1050
86 IFA$="T"THENGOSUB1060
88 IFA$="G"THENGOSUB1070
90 IFA$="Y"THENGOSUB1080
92 IFA$="H"THENGOSUB1090
94 IFA$="U"THENGOSUB1100
96 IFA$="J"THENGOSUB1110
98 IFA$="K"THENGOSUB1120
100 IFA$="O"THENGOSUB1130
102 IFA$="L"THENGOSUB1140
104 IFA$="P"THENGOSUB1150
106 IFA$=":"THENGOSUB1160
108 IFA$=" "THENGOSUB1170
110 IFA$=CHR$(133)THENGOSUB1180
112 IFA$=CHR$(134)THENGOSUB1190
114 IFA$=CHR$(135)THENGOSUB1200
120 GOTO74
1000 POKES,189
1001 GOSUB2000
1002 POKECA+265,81:POKECO+265,0:K=265:J=32
1005 RETURN
1010 POKES,193
1011 GOSUB2000
1012 POKECA+156,81:POKECO+156,0:K=156:J=160
1015 RETURN
1020 POKES,196
1021 GOSUB2000
1022 POKECA+267,81:POKECO+267,0:K=267:J=32
1025 RETURN
1030 POKES,199
1031 GOSUB2000
1032 POKECA+158,81:POKECO+158,0:K=158:J=160
1035 RETURN
1040 POKES,203
1041 GOSUB2000
1042 POKECA+269,81:POKECO+269,0:K=269:J=32
1045 RETURN
1050 POKES,205
1051 GOSUB2000
1052 POKECA+271,81:POKECO+271,0:K=271:J=32
1055 RETURN
1060 POKES,208
1061 GOSUB2000
1062 POKECA+162,81:POKECO+162,0:K=162:J=160
1065 RETURN
```

```
1070 POKES,211
1071 GOSUB2000
1072 POKECA+273,81:POKECO+273,0:K=273:J=32
1075 RETURN
1080 POKES,213
1081 GOSUB2000
1082 POKECA+164,81:POKECO+164,0:K=164:J=160
1085 RETURN
1090 POKES,216
1091 GOSUB2000
1092 POKECA+275,81:POKECO+275,0:K=275:J=32
1095 RETURN
1100 POKES,218
1101 GOSUB2000
1102 POKECA+166,81:POKECO+166,0:K=166:J=160
1105 RETURN
1110 POKES,220
1111 GOSUB2000
1112 POKECA+277,81:POKECO+277,0:K=277:J=32
1115 RETURN
1120 POKES,222
1121 GOSUB2000
1122 POKECA+279,81:POKECO+279,0:K=279:J=32
1125 RETURN
1130 POKES,224
1131 GOSUB2000
1132 POKECA+170,81:POKECO+170,0:K=170:J=160
1135 RETURN
1140 POKES,225
1141 GOSUB2000
1142 POKECA+281,81:POKECO+281,0:K=281:J=32
1145 RETURN
1150 POKES,227
1151 GOSUB2000
1152 POKECA+172,81:POKECO+172,0:K=172:J=160
1155 RETURN
1160 POKES,229
1161 GOSUB2000
1162 POKECA+283,81:POKECO+283,0:K=283:J=32
1165 RETURN
1170 POKES,0:RETURN
1180 POKES,0:S=36876:RETURN
1190 POKES,0:S=36875:RETURN
1200 POKES,0:S=36874:RETURN
2000 POKECA+K,J:POKECO+K,0:RETURN
3000 DATA66,72,80,86,88,94,102,108,110,116,124,130,132,138
3001 DATA146,152,154,160,168,174,176,182,190,-1
3002 DATA68,70,74,76,78,82,84,90,92,96,98,100,104,106,112,114
3003 DATA118,120,122,126,128,134,136,140,142,144,148,150,156
3004 DATA158,162,164,166,170,172,178,180,184,186,188,192,194,-1

READY.
```

Controllo del cursore e videogiochi

Un sottoprogramma utile per generare videogames.

SCRIVIAMO malvolentieri articoli sui videogiochi perchè rappresentano il prodotto deteriore di una malsana società consumistica, ma soprattutto per via che i nostri redattori non sono mai riusciti a conseguire un punteggio dignitoso (ma come fanno certi dannati ragazzini a raggiungere "score" incredibili?).

Scherzi a parte profitteremo anche stavolta per fare un po' di didattica, consentendo ai principianti di capire in che modo sia possibile, in un videogioco, spostare lungo lo schermo un... qualcosa (pedina o cursore).

Tabella 1

Indirizzi delle locazioni richiamate nell'articolo.

Config. Locazioni	Inespanso	+3K	+8K in poi
Basic	4096-7679	1024-7679	4608 in poi
Schermo	7680-8191	7680-8191	4096-4607
Colore	38400-38911	38400-38911	37888-38399

Un cenno alle locazioni di schermo

Ad ogni carattere visualizzabile sullo schermo, corrisponde una locazione di memoria, nella quale è possibile "scrivere" il carattere stesso. Ne deriva che le locazioni di schermo sono 506, valore questo, dato dal prodotto di 22 colonne per 23 righe. In effetti la quantità di memoria RAM necessaria per la gestione dello schermo è esattamente il doppio di 506. Associate alle locazioni di schermo sono, infatti, da considerare altre 506 locazioni, dette, per brevità, "di colore".
Il carattere di schermo come vedremo tra breve, deve essere "colorato" affinchè si renda visibile. Inoltre i cosiddetti indirizzi delle locazioni di schermo e colore (rappresentate da due blocchi di 506 locazioni contigue RAM) cambiano a seconda della eventuale presenza di espansioni di memoria RAM (vedi tabella 1).

Da un attento esame ci accorgiamo anzitutto che la differenza dei valori estremi di schermo (es. 8191-7680) risulta essere 511 e non 506. Il motivo di questa diversità esula dallo scopo del presente articolo e sarà riproposto in altra occasione. Le considerazioni che seguono, terranno conto pertanto dei primi 506 byte di schermo e, ovviamente, dei corrispondenti 506 byte di colore. Dalla tabella si nota, inoltre, che le locazioni di schermo sono in coda a quelle dedicate ai programmi Basic nei casi di Vic 20 in versione base (inespanso) ed in quella con la scheda da 3K, mentre sono ubicate prima dell'area Basic, nel caso di Vic dotati di cartuccia da 8K oppure 16K di RAM. Il motivo del cambio di indirizzo, che è realizzato in modo completamente automatico al-

l'atto dell'accensione del computer, è da ricercarsi, come vedremo in altri articoli, nell'ottimizzazione della memoria disponibile.

Prima fase per la comprensione del listato

Eseguite nell'ordine i passi seguenti:
1/ accendete il computer dotato, oppure no, di un'espansione di memoria.
2/ Cancellate lo schermo (tasti SHIFT + CLR/HOME).

3/ Premete due volte il tasto Return.
4a/ Digitate:

```
A=7680: B=38400
```

se non avete espansioni o se avete inserito quella da 3K.
4b/ digitate:

```
A=4096 : B=37888
```

nel caso in cui abbiate inserito l'espansione RAM da 8 oppure 16K.
5/ Digitate:

```
POKE A,0: POKE B,0
```

Se avete seguito (ALLA LETTERA) le semplici indicazioni dovreste a questo punto vedere nell'angolo in alto a sinistra il carattere @ (gli inglesi lo leggono, at mentre in Italia è invalso l'uso di chiamarlo "chiocciola"), scritto in nero.
6/ Digitate:

```
POKE A+1,0
```

Questa operazione dovrebbe, a rigor di logica, fare apparire lo stesso carattere @ nella locazione successiva a quella numerica con A (seconda cella in alto a sinistra). Notiamo con sorpresa che ciò non accade. Ma se noi...
7/... saliamo in alto col cursore (tasto HOME) e lo posizioniamo nella seconda cella in alto a sinistra noteremo che il carattere @ viene evidenziato ad ogni lampeggio del cursore.
8/ Torniamo, grazie al tasto CRSR (quella per gli spostamenti verticali) sul rigo del comando digitato al punto 6 e modifichiamolo in:

```
POKE A+1,0: POKE B+1,0
```

Compare, finalmente, la seconda chiocciolina.

L'esperimento appena visto dimostra che:
- è possibile fare apparire in qualsiasi locazione dello schermo, a patto ovviamente di calcolarne l'indirizzo, uno qualsiasi dei 256 caratteri Commodore (maiuscole, minuscole, caratteri grafici, caratteri reversati);
- è indispensabile, dopo aver trascritto il carattere nella locazione di schermo desiderata, colorare la cella stessa, servendosi del comando POKE che agisce sulla cosiddetta mappa del colore schermo. Si fa notare che se conferiamo al carattere visualizzato lo stesso colore del fondo, il carattere stesso risulta invisibile benchè... presente. Esempio: rosso su rosso, verde su verde, ecc.
- quando si digita un comando del tipo:

```
PRINT "PROVA"
```

il computer non solo provvede a far apparire i caratteri desiderati, ma colora automaticamente le celle di memoria destinate ad ospitare i caratteri stessi. Ricorrendo, invece, ad operazioni di POKE che interessano l'area di schermo è come se perdessimo tale automatismo, e quindi siamo costretti, per evidenziare un carattere, a colorarlo mediante un secondo comando di POKE agente sull'area colore;
- gli argomenti delle POKE di schermo non corrispondono sempre agli argomenti delle istruzioni PRINT CHR$(X). Esempio: la battitura di

```
POKE A,80: POKE B,0
```

(intendendo con A e B le locazioni di schermo e del colore) porta alla visualizzazione di un carattere diverso da:

```
PRINT CHR$(80)
```

- ricorrendo a PRINT non ci si deve preoccupare di indicare le locazioni di schermo che saranno interessate, mentre, utilizzando i POKE di schermo e di colore, è indispensabile controllare attentamente gli indirizzi prima di eseguire il comando.

Una locazione interessante: la 203

Tutte le volte che premiamo un tasto del Vic 20, il Sistema Operativo (O.S.) del computer

provvede a individuarlo, a codificarlo con un valore intero compreso tra 0 e 255 e a memorizzarlo nella locazione di memoria 203. Per renderci conto di quanto asserito, facciamo girare a parte il microprogramma riportato nella riga 150 del listato pubblicato:

```
150 PRINT PEEK(203):
GOTO 150
```

Non appena digitiamo RUN noteremo che sullo schermo compare in continuazione il numero 64. Questo è infatti, il codice del Vic che indica all'O.S. che nessun tasto viene premuto in quel momento. Proviamo ora a premere un tasto qualunque e teniamolo pigiato: il numero 64 sarà immediatamente sostituito da un nuovo valore che rappresenta il codice relativo al tasto premuto. La riga 150 serve dunque per esaminare i codici di ciascun tasto del Vic.

Che cosa fa il programma

Il programma, per lo scopo didattico per cui è stato scritto, si limita a far spostare, sullo schermo del Vic, ad una velocità stabilita, un carattere controllato dalla pressione di alcuni tasti:

Y	movimento	verso	l'alto
B	"	"	il basso
G	"	"	sinistra
J	"	"	destra

Il carattere che viene spostato sullo schermo non è sempre lo stesso. Infatti nei movimenti verso il basso assume la forma di una "V", in quelli verso l'alto di "↑", verso destra diventa ">" e verso sinistra prende la forma di "<". Ho pensato, in tal modo, di conferire maggior... realismo al movimento e di dare un piccolo suggerimento per sofisticazioni da introdurre.

Come funziona

Volendo, il carattere che si muove, può lasciare dietro a sè una scia. Se infatti alla domanda CARATTERE? (Riga 180) rispondiamo con 32, il carattere si muoverà lungo lo schermo senza lasciare apparentemente alcuna traccia (in realtà viene invece trascritto il carattere di spazio bianco). Al valore 160 corrisponderà una scia nera e ad altri valori, compresi ovviamente tra 0 e 255, corrisponderanno i vari caratteri tipici del Vic, da individuare grazie alla riga 150.

Riga 200-210. A seconda della espansione posseduta e controllando la tabella 1, il lettore trascriverà la riga 200 (cancellando REM) oppure la 210.

Riga 220. Calcolo dei blocchi di memoria RAM dedicati allo schermo e al colore.

Riga 230. Invece di provvedere, nel seguito del programma, a colorare una cella video, dopo aver depositato un carattere, si è preferito colorare ciascuna cella dello schermo fin dall'inizio, in modo da snellire il programma stesso.

Righe 240-290. Esame dell'eventuale tasto premuto. In caso di pressione di tasti diversi da B,Y,G,J o di nessun tasto, si ritorna continuamente (riga 290) alla riga 240.

Righe 310-320. Spostamento in basso del carattere. Prima di spostarlo è però necessario controllare che non si esca dalla mappa del video (riga 310). Se infatti il carattere, al momento della pressione del tasto B, è già presente sull'ultima riga, non ha senso dare il comando di un'ulteriore discesa.

Righe 340-350. Controllo salita carattere.

Righe 370-380. Spostamento a destra. La cella interessata è individuata dal valore S2 (numero di cella contenente il carattere PRIMA della pressione di J) incrementato di un'unità; il carattere presente nella cella video, prima della pressione di J, deve essere sostituito dal carattere scelto alla riga 180. Infine (riga 380) il valore di S2 viene "aggiornato" per le successive elaborazioni.

Righe 400-410. Spostamento a sinistra.

Riga 440. La variabile LO, definita alla riga 170, consente di introdurre un ritardo nello spostamento del carattere tra una cella e la successiva. Se infatti alla domanda LIVELLO? rispondiamo col valore nullo, la velocità di spostamento è piuttosto elevata ed è necessaria una certa prontezza di riflessi per il controllo del cursore.

Quello proposto, come già detto, deve essere considerato come sottoprogramma, magari da sofisticare, in listati di videogiochi, in cui è necessario spostare una pedina sullo schermo. Un piccolo suggerimento prima di concludere: dopo aver trascritto il programma dalla rivista, inserite la riga 190, sostituite la 240 con l'altra riportata in fondo al listato e... buon divertimento! •

```
100.REM *** CONOSCERE MEGLIO IL VIC 20 ***
110 REM ***                          ***
120 REM ***  ROUTINE FONDAMENTALI PER  ***
130 REM ***  LA GENERAZIONE DI GIOCHI  ***
140 :
150 REM PRINT PEEK (203): GOTO 150: ESAME CODICE CARATTERE
160 :
170 INPUT"LIVELLO (0-1000)"; LOOP: REM DIFFICOLTA' = RITARDO
180 INPUT"CARATTERE (0-255)";CA: IF CA>255 THEN 180: REM CARATTERE... "SCIA"
190 :
200 REM S1 = 7680: C1 = 38400: REM VIC INESPANSO
210 S1 = 4096: C1 = 37888:     REM VIC ESPANSO (DA 8K IN SU)
220 S2=S1: S3=S1+22*23-1: C2=C1+22*23-1: PRINT "♥"
230 FOR X=C1 TO C2 :POKE X,0: NEXT: REM COLORAZIONE FONDO SCHERMO
240 X = PEEK (203): REM ESAME TASTO EVENTUALMENTE PREMUTO
250 IF X=35 THEN GOSUB 310: GOSUB 440: GOTO 240: REM TASTO B
260 IF X=11 THEN GOSUB 340: GOSUB 440: GOTO 240: REM TASTO Y
270 IF X=19 THEN GOSUB 400: GOSUB 440: GOTO 240: REM TASTO G
280 IF X=20 THEN GOSUB 370: GOSUB 440: GOTO 240: REM TASTO J
290 GOTO 240: REM A. DE SIMONE SOFTWARE '83
295 :
300 REM MOVIMENTO VERSO IL BASSO. 22 = CARATTERE V
310 IF S2+22 > S3 THEN RETURN
320 POKE S2+22,22: POKE S2,CA :S2 = S2+22: RETURN
325 :
330 REM MOVIMENTO IN ALTO. 30 = CARATTERE ↑
340 IF S2-22 < S1 THEN RETURN
350 POKE S2-22,30: POKE S2,CA :S2 = S2-22: RETURN
355 :
360 REM MOVIMENTO A DESTRA. 62 = CARATTERE >
370 IF S2+1 > S3 THEN RETURN
380 POKE S2+1,62: POKE S2,CA :S2 = S2+1: RETURN
385 :
390 REM MOVIMENTO A SINISTRA. 60 = CARATTERE <
400 IF S2-1 < S1 THEN RETURN
410 POKE S2-1,60: POKE S2,CA: S2 = S2-1: RETURN
420 :
430 REM SOTTOPROGRAMMA DI RITARDO
440 FOR I=1 TO LOOP: NEXT: RETURN

READY.

190 A(1)=35: A(2)=11: A(3)=19: A(4)=20

240 X = A(INT(RND(J) * 4))
```

DAL N°1:

Speciale, specialissimo!
Invece dei soliti videogiochi prova VIC 20, e guarda quante cose fa in più!

1. VIC 20 ha una valanga di videogiochi, uno più bello dell'altro, uno più nuovo dell'altro.
2. Ma VIC 20 è un computer e fa molto di più.
3. Lo usi per la scuola, o per la casa, o per la professione. Ci sono, pronti pronti, un mucchio di programmi. Metti le cassette e via con cose utili.
4. Puoi imparare il BASIC, la lingua del futuro (ed è facile facile imparare a programmare).
5. Nel mondo sono stati venduti più di un milione di VIC, a gente sveglia, quelli del 2000.

Ed ora passo al Pet

Come leggere col Pet (4000 8000) i programmi scritti con i computer Vic 20 e Commodore 64.

DOVREBBE essere noto che i computer dotati di Basic Microsoft, e in particolare i Commodore, utilizzano aree di memoria ben definite per deporre i programmi dell'utente. L'indirizzo di partenza dei listati è contenuto in due locazioni di memoria che rappresentano, rispettivamente, la parte bassa e quella alta dell'indirizzo stesso.

Negli elaboratori PET (qualsiasi serie) e nel VIC 20, dotato, però, dell'espansione di memoria di 3K di RAM, i programmi Basic vengono posizionati a partire dalla locazione 1024; nel Vic senza espansioni, dalla locazione 4096, nel Vic con espansioni da 8 e più K, dalla locazione 4608 e nel Commodore 64, da 2048.

Quando impartiamo l'ordine SAVE, l'inizio del file-programma contiene l'indirizzo di partenza in cui è allocato il programma che sta per essere registrato. Quando invece diamo il comando LOAD, il comportamento dei PET si diversifica da quello dei VIC (20 e 64).

Nel primo caso, infatti, il PET carica, *sempre e comunque,* il programma considerando l'indirizzo posto in testa al file.

Nei casi del Vic 20 e del Commodore 64, invece, il programma viene posizionato automaticamente a partire dalla locazione in cui il computer stesso, nella particolare configurazione di memoria adoperato, allocherebbe un programma se questo fosse digitato normamente da tastiera.

In altre parole, un programma registrato in un momento in cui il Vic 20 era senza espansioni di memoria, (locazione inizio Basic = 4096), verrà posizionato, se riletto con espansione di memoria da 8K, a partire da 4608.

L'utente non si accorge affatto del cambiamento, proprio grazie all'automatismo della procedura.

Sembrerebbe, pertanto, che non è possibile caricare su PET un programma scritto dai suoi... fratelli minori. Naturalmente, siamo qui per illustrarvi la possibilità di riuscire nell'intento, a patto di osservare *alla lettera* la procedura qui di seguito riportata [(R) = battere il tasto di Return]:

1/ caricare sul PET il programma in oggetto col "solito" LOAD;

2/ quando l'operazione è completata, *non listate* il programma per nessun motivo;

3/ entrate in monitor digitando: SYS (1024) (R);

4/ visualizzate il rigo di otto byte battendo: M 0400 0401 (R);

5/ ignorate il contenuto delle otto locazioni di memoria visualizzate dopo la fase 4/;

6/ servendosi dei tasti di controllo cursore, modificate le locazioni specificate come segue:

- caso di programma registrato con Vic 20 senza alcuna espansione di memoria:

0400 : 00 01 10 00 00 00 00 00 (R);

- Caso di Vic 20 con espansione da 8 o più K RAM:

0400 : 00 01 12 00 00 00 00 00 (R);

- caso di Commodore 64:

0400 : 00 01 08 00 00 00 00 00 (R);

7/ "uscire" dal monitor digitando "X" (R);

8/ *non* chiedere il listato in questo momento: il PET andrebbe in Tilt;

9/ digitare:

0 (R).

Comportandovi, cioè, come se voleste cancellare, da un normalissimo programma Basic, la linea numerata con zero.

A questo punto il programma è posizionato correttamente in memoria: potete listarlo, usarlo, modificarlo, ecc. Nel caso abbiate un programma di cui non si conosce il computer di provenienza, l'unico modo è quello di procedere per tentativi fino a che non appare il listato. Si tenga presente che, come già detto, se il programma proviene da un Vic 20, dotato di 3K RAM, non è necessaria alcuna modifica. Se, invece, nonostante il procedimento suggerito, non appare il listato o il PET si "impianta" ad ogni tentativo, vorrà dire che il programma in oggetto è scritto non in Basic, ma in Linguaggio Macchina, oppure, è protetto. In questo caso non c'è nulla da fare... ▪

DAL N°1: COM

Mai un grande perso

Quest'anno, fatti un regalo intelligente: un computer dalle caratteristiche incredibili. Vediamole.

1. Commodore 64 è potente, sofisticato, professionale.

2. Ha una vastissima gamma di programmi già pronti, lo usi nella professione, a casa, a scuola, nella ricerca scientifica, con facilità e totale affidabilità.

3. Ha un'incredibile memoria (64 K), un sintetizzatore sonoro professionale, produce effetti tridimensionali.

4. Ti diverti perchè è anche un sofisticato videogioco.

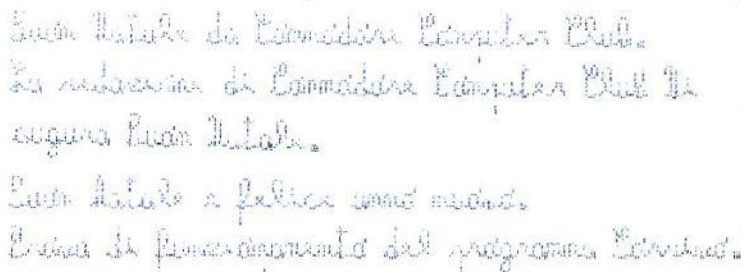
5. Con Commodore 64 entri nel futuro, tasto dopo tasto.

6. Commodore 64 oggi lo puoi avere a prezzo davvero speciale:

Stampa in corsivo

Una scrittura "personalizzata" con qualsiasi sistema Commodore.

Di Bellù Giovanni

Esempio di stampa ottenibile col programma proposto.

IL programma, che gira su configurazione Pet + stampante 4022 (o 3022) e di cui viene data anche la versione per il Vic con almeno l'espansione di 8K, come già indica il nome, serve a fare scrivere il computer in corsivo, cioè con quel tipo di scrittura che noi abitualmente usiamo quando scriviamo a mano.

Come si usa

All'inizio bisogna attendere un po' - circa mezzo minuto - per permettere al computer di leggere ed elaborare tutti quei numeri (circa 1500!) che sono scritti nei DATA. Quindi, appare sul video la scritta "INTRODUCI IL TESTO", ed una linea lunga 79 caratteri - questa è la lunghezza massima di una riga del testo da fare stampare.

L'input del testo avviene mediante l'uso dell'istruzione GET, il che si è reso necessario per controllare i dati introdotti; e che per il resto è uguale all'input: si può, cioè, usare il tasto DELETE, e bisogna premere RETURN alla fine della riga.

In seguito si può introdurre un'altra riga di testo, fino ad un massimo di 50 (Questo valore si può cambiare modificando il valore della DIMB$(50) di riga 270).

Per indicare che si è finito di introdurre il testo, bisogna premere "*" all'inizio di una nuova riga. Se questo carattere viene usato non in prima posizione, esso viene accettato e naturalmente stampato. A questo punto comincia la stampa.

Per quanto riguarda i caratteri che si possono usare, essi sono:

- tutte le lettere dell'alfabeto latino sia minuscole che maiuscole (minuscole shiftate);
- tutti i caratteri numerici;
- i caratteri di punteggiatura: .,;:? !
- altri caratteri che in ascii vanno da 32 (spazio) al 64 (a commerciale).

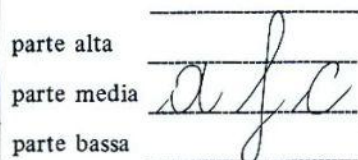
Tutti gli altri caratteri non vengono accettati. Una volta copiato il listato potrete scrivere tutto quello che volete e, dato che siamo in clima natalizio, pòtete mandare le vostre lettere di auguri scritte dal Computer...

Come funziona

Tutti quei numeri troppi?) nei DATA servono per la formazione dei caratteri speciali che si possono formare usando una del-

le caratteristiche della stampante Commodore.

Il metodo usato per la formazione dei caratteri consiste nello scomporre ogni lettera in tre parti: alta, media, bassa come in figura:

parte alta
parte media
parte bassa

Per questa ragione per ogni lettera ci sono tre caratteri speciali per cui ci sono 18 dati per ciascuna lettera (6 per carattere per tre caratteri).

Nella fase di stampa, per ogni riga di testo, verranno stampate tutte le parti "alte" delle lettere, poi quelle "medie", ed infine quelle "basse", per cui ogni riga di testo occupa tre righe di stampa.

La versione VIC

Il procedimento di formazione dei caratteri è il medesimo, con l'unica differenza che, mentre nel Pet la formazione dei 6 numeri per ogni carattere speciale avviene tenendo conto delle potenze di 2 a partire dal 6 fino allo 0, nel VIC è esattamente l'opposto, come in figura.

Pet: 2 elev. 6	Vic: 2 elev.0
2 elev. 5	2 elec. 1
...	...
2 elev. 0	2 elev. 6

Per cui si è reso necessario, per non rifare la mappa caratteri, fare una "inversione". Questo avviene nelle linee di programma da 10000 a 10012, in cui il numero viene trasformato in binario, quindi, "ribaltato" e ritrasformato in decimale.

Altre modifiche riguardano la parte di stampa:
+ 250 per i diversi OPEN e istruzioni grafiche;
+ 300 per la formazione dei caratteri su VIC bisogna sommare 128 al numero nei DATA;
+ 1280 e 1290 per le istruzioni PRINT#.

Per il resto la versione VIC è esattamente identica a quella per il Pet.

Programma "crea"

Questo programmino si è rivelato molto utile per la definizione dei caratteri.

All'inizio chiede quale carattere dobbiamo formare, poi sullo schermo appaiono tre rettangoli di 6*7, uno sopra l'altro, che rappresentano le tre parti in cui viene divisa la lettera (vedi prima). In fondo allo schermo appare il cursore lampeggiante (in input).

A questo punto bisogna solamente salire con gli appositi tasti di controllo del cursore e mettere il carattere "Q shiftato" nelle posizioni in cui sulla stampante deve venire stampato un punto.

Finito di formare il carattere, come meglio preferiamo, basta premere RETURN su una linea in cui vi siano dei caratteri (non importa quali, e questo lo potrete capire facilmente analizzando il programma o, meglio ancora, facendolo girare.

Quindi il computer analizza i tre rettangoli di prima e automaticamente visualizza sullo schermo il numero di linea corrispondente a quel carattere seguito da un DATA che contiene i 18 dati trovati, ed infine, un'istruzione REM seguita dal carattere in questione, e sotto a questa linea c'è la scritta RUN.

A questo punto dobbiamo premere RETURN 2 volte; una per memorizzare la linea che contiene i dati del carattere in questione, l'altra per ritornare nel programma.

Nel frattempo, il carattere che abbiamo formato viene stampato, per cui possiamo verificare subito se c'è qualche cosa da modificare o se va bene. ▪

Bellù Giovanni

```
C   O   R   S   I   V   O
Di Bellu' Giovanni - Nov. 1983 -

100 REM ************************************
110 REM **        C O R S I V O           **
120 REM ************************************
```

```
130 REM ** VERSIONE PET+4022 (3022) **
140 REM ******************************
150 REM **    BY BELLU' GIOVANNI    **
160 REM ******************************
170 REM **    VIA GIARDINI N. 20    **
180 REM ******************************
190 REM **      20038  SEREGNO      **
200 REM ******************************
210 REM **  TELEFONO: 0362/239580   **
220 REM ******************************
230 REM **     - NOVEMBRE 1983 -    **


240 REM ******************************
250 OPEN6,4,6:PRINT#6,CHR$(22):OPEN3,4:OPEN5,4,5:POKE59468,14
260 PRINT"□▣   ♠TTENDI  STO  MEMORIZZANDO I  DATI   "
270 DIMA$(85,2),A(18),B$(50)
280 FORK=1TO85
290 FORJ=1TO18:READA(J):NEXTJ
300 FORH=0TO2:FORJ=1TO6:A$(K,H)=A$(K,H)+CHR$(A(J+6*H)):NEXTJ,H,K
310 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:REMSPACE
320 DATA0,0,127,127,0,0,0,0,123,123,0,0,0,0,0,0,0,0:REM
330 DATA8,16,96,8,16,96,0,0,0,0,0,0,0,0,0,0,0:REM"
340 DATA0,0,0,0,0,0,20,127,20,127,20,0,0,0,0,0,0:REM#

350 DATA96,24,4,63,68,56,6,41,41,127,36,34,0,0,0,0,0,0:REM$
360 DATA1,1,1,0,0,1,99,36,104,23,37,71,0,0,0,0,0,0:REM%
370 DATA1,2,2,2,1,0,102,25,9,21,114,5,0,0,0,0,0,0:REM&
380 DATA0,0,6,28,112,0,0,0,0,0,0,0,0,0,0,0,0:REM'
390 DATA0,31,32,64,64,64,0,124,2,1,1,1,0,0,0,0,0,0:REM(
400 DATA64,64,64,32,31,0,1,1,1,2,124,0,0,0,0,0,0:REM)
410 DATA0,1,3,3,1,0,0,64,96,96,64,0,0,0,0,0,0,0:REM*
420 DATA1,1,7,7,1,1,64,64,112,112,64,64,0,0,0,0,0,0:REM+
430 DATA0,0,0,0,0,0,0,3,3,0,0,0,4,24,96,0,0:REM,
440 DATA1,1,1,1,1,64,64,64,64,64,64,0,0,0,0,0,0:REM-
450 DATA0,0,0,0,0,0,0,7,7,7,0,0,0,0,0,0,0:REM.
460 DATA1,1,25,25,1,1,64,64,76,76,64,64,0,0,0,0,0,0:REM/
470 DATA31,32,64,64,32,31,124,2,1,1,2,124,0,0,0,0,0,0:REM0
480 DATA8,16,32,64,127,0,0,1,1,1,127,1,0,0,0,0,0,0:REM1
490 DATA48,64,64,65,66,60,28,34,65,1,1,1,0,0,0,0,0,0:REM2
500 DATA32,65,65,65,62,0,2,65,65,65,62,0,0,0,0,0,0,0:REM3
510 DATA0,0,1,6,24,96,0,60,68,4,31,4,0,0,0,0,0,0:REM4
520 DATA0,127,65,65,64,96,2,1,1,1,126,0,0,0,0,0,0,0:REM5
530 DATA7,8,16,32,64,64,124,2,33,33,33,30,0,0,0,0,0,0:REM6
540 DATA112,64,64,65,78,112,0,67,92,96,64,64,0,0,0,0,0,0:REM7
550 DATA28,34,65,65,34,28,60,66,1,1,66,60,0,0,0,0,0,0:REM8
560 DATA60,66,66,66,63,0,6,1,1,2,124,0,0,0,0,0,0,0:REM9
570 DATA0,0,0,0,0,0,0,99,99,0,0,0,0,0,0,0,0:REM:
580 DATA0,0,0,0,0,0,0,51,51,0,0,0,4,24,96,0,0:REM;
590 DATA0,0,0,1,1,24,24,36,102,67,1,0,0,0,0,0,0:REM<
600 DATA6,6,6,6,6,6,48,48,48,48,48,48,0,0,0,0,0,0:REM=
```

```
610 DATA1,1,0,0,0,0,1,67,102,36,24,24,0,0,0,0,0,0:REM>
620 DATA24,32,64,65,66,60,0,0,125,0,0,0,0,0,0,0,0:REM?
630 DATA0,0,0,0,0,0,62,65,93,85,84,60,0,0,0,0,0,0:REM@
640 DATA0,0,0,0,0,0,2,62,65,65,62,7,0,0,0,0,0,0:REM A
650 DATA0,31,32,32,31,0,2,124,31,33,78,10,0,0,0,0,0,0:REM B
660 DATA0,0,0,0,0,0,2,28,34,65,65,2,0,0,0,0,0,0:REM C
670 DATA0,0,0,0,127,0,2,30,33,33,127,6,0,0,0,0,0,0:REM D
680 DATA0,0,0,0,0,0,2,52,72,52,2,2,0,0,0,0,0,0:REM E
690 DATA0,31,32,32,31,0,2,127,13,50,66,2,0,127,0,0,0,0:REM F
700 DATA0,0,0,0,0,0,2,62,65,65,63,3,0,28,34,66,126,0:REM G
710 DATA0,31,32,32,31,0,2,127,10,52,98,3,0,0,0,0,0,0:REM H
720 DATA0,0,2,0,0,0,2,4,120,12,4,2,0,0,0,0,0,0:REM I
730 DATA0;0,0,0,1,0,3,0,0,0,63,1,0,112,8,8,112,0:REMJ
740 DATA0,0,0,0,0,0,2,127,8,20,34,66,0,0,0,0,0,0:REMK
750 DATA0,31,32,32,31,0,2,4,120,24,100,2,0,0,0,0,0,0:REM L
760 DATA0,0,0,0,0,0,126,64,126,64,124,2,0,0,0,0,0,0:REM M
770 DATA0,0,0,0,0,0,62,64,62,64,60,2,0,0,0,0,0,0:REM N
780 DATA0,0,0,0,1,6,2,62,65,65,126,2,0,0,0,0,0,0:REM O
790 DATA0,0,0,0,0,0,2,12,16,96,127,2,0,0,0,0,127,0:REM P
800 DATA0,0,0,0,0,0,2,30,33,65,127,2,0,0,0,0,127,0:REM Q
810 DATA0,0,0,0,0,0,2,12,112,64,120,6,0,0,0,0,0,0:REM R
820 DATA0,0,0,0,0,0,2,12,48,66,66,62,0,0,0,0,0,0:REM S
830 DATA0,16,127,16,16,0,2,4,127,1,2,2,0,0,0,0,0,0:REM T
840 DATA0,0,0,0,0,0,2,126,1,1,126,2,0,0,0,0,0,0:REMU
850 DATA0,0,0,0,0,0,30,16,15,1,14,18,0,0,0,0,0,0:REMV
860 DATA0,0;0,0,0,0,2,127,2,4,2,127,0,0,0,0,0,0:REMW
870 DATA0,0,0,0,0,0,99,20,8,20,99,2,0,0,0,0,0,0:REMX
880 DATA0,0,0,0,0,0,2,126,1,2,127,2,0,3,12,48,64,0:REMY
890 DATA0,0,0,0,0,0,14,112,70,74,82,98,0,0,0,0,0,0:REM Z
900 DATA0,31,32,32,31,0,2,126,1,1,126,1,0,0,0,0,0,0:REM A
910 DATA0,127,65,66,60,0,2,127,1,65,62,2,0,0,0,0,0,0:REM B
920 DATA96,27,4,58,66,60,2,126,1,1,3,2,0,0,0,0,0,0:REM C
930 DATA0,127,64,48,15,0,2,127,1,2,124,2,0,0,0,0,0,0:REM D
940 DATA64,48,15,52,36,24,2,6,73,41,17,18,0,0,0,0,0,0:REM E
950 DATA120,64,67,76,112,64,66,71,97,81,79,2,0,0,0,0,0,0:REM F
960 DATA64,51,28,40,72,112,2,124,3,1,15,3,6,9,17,34,124,0:REM G
970 DATA0,127,1,1,127,0,2,126,0,0,126,2,0,0,0,0,0,0:REM H
980 DATA112,64,64,67,124,0,2,14,1,97,30,2,0,0,0,0,0,0:REM I
990 DATA0,0,0,0,127,0,2,1,1,1,126,0,0,0,0,0,0,0:REM J
1000 DATA0,127,3,4,8,112,2,127,0,64,48,15,0,0,0,0,0,0:REM K
1010 DATA64,56,7,60,68,56,2,5,127,8,4,2,0,0,0,0,0,0:REM L
1020 DATA127,64,127,64,127,0,126,0,126,0,120,6,0,0,0,0,0,0:REM M
1030 DATA48,64,63,64,127,0,2,4,126,0,124,2,0,0,0,0,0,0:REM N
1040 DATA0,63,64,64,63,64,2,126,1,1,126,0,0,0,0,0,0,0:REM O
1050 DATA0,127,65,64,63,0,2,127,65,65,65,2,0,0,0,0,0,0:REM P
1060 DATA0,63,64,64,64,63,2,126,1,9,5,122,0,0,0,0,0,0:REM Q
1070 DATA0,127,64,65,62,0,2,127,64,32,28,2,0,0,0,0,0,0:REM R
1080 DATA96,16,60,75,72,48,6,9,1,65,57,6,0,0,0,0,0,0:REM S
1090 DATA96,64,65,70,120,0,2,60,70,1,1,6,0,0,0,0,0,0:REM T
1100 DATA0,127,0,0,127,0,2,126,1,1,126,2,0,0,0,0,0,0:REM U
1110 DATA56,64,127,0,31,32,2,2,126,1,127,2,0,0,0,0,0,0:REM V
1120 DATA0,127,0,0,0;127,2,127,6,56,6,127,0,0,0,0,0,0:REM W
```

```
1130 DATA112,14,1,14,112,0,3,28,96,28,2,2,0,0,0,6,0,0:REM X
1140 DATA64,56,7,0,127,0,2,1,2,92,100,4,0,0,0,0,0,0:REM Y
1150 DATA96,64,64,67,76,112,67,77,113,65,65,7,0,0,0,0,0,0:REM Z
1160 PRINT"█"
1170 PRINT"███ INTRODUCI IL TESTO "
1180 FORK=0TO999:NEXT
1190 GOSUB1360
1200 FORI=0TOW-1
1210 FORJ=0TO2
1220 FORK=1TOLEN(B$(I))
1230 C$=MID$(B$(I),K,1)
1240 A=ASC(C$)
1250 IFA>31ANDA<91THENA=A-31:GOTO1280
1260 IFA>192ANDA<219THENA=A-192+59:GOTO1280
1270 PRINT"██ ERRORE ":END
1280 PRINT#5,A$(A,J):PRINT#3,TAB(K)CHR$(254)CHR$(141);:NEXT
1290 PRINT#5,A$(1,1):PRINT#3,TAB(K)CHR$(254):NEXT
1300 A$="":NEXT
1310 PRINT"██ FINE "
1320 PRINT"███ ALTRO TESTO ?"
1330 GETA$:IFA$<>"S"ANDA$<>"N"THEN1330
1340 IFA$="N"THENEND
1350 W=0:GOTO1160
1360 P3=78:P4=2
1370 P$=""
1380 PRINT
1390 FORQ=1TOP3:PRINT"_";:NEXT:PRINT"| ";
1400 FORQ=1TOP3+1:PRINT"█";:NEXT
1410 REM
1420 GETA$:IFA$<>""THEN1450
1430 FORJ=1TO20:PRINT"_█";:NEXT:GETA$:IFA$<>""THEN1450
1440 FORJ=1TO10:PRINT"█ ██";:NEXT:GOTO1420
1450 IFA$=CHR$(13)THENPRINT"_█";:GOTO1610
1460 IFASC(A$)=34THENPRINTA$A$CHR$(20)CHR$(148)"_█";:P$=P$+A$:GOTO141
1470 IFP$=""ANDA$="*"THENRETURN
1480 IFA$<>CHR$(20)GOTO1540
1490 PRINT"_█";
1500 IFP$=""GOTO1410
1510 IFLEN(P$)=1THENP$="":GOTO1530
1520 P$=LEFT$(P$,LEN(P$)-1)
1530 A$="█_█":GOTO1590
1540 IFA$=CHR$(160)THENA$=" "
1550 IFASC(A$)>90ANDASC(A$)<193THEN1410
1560 IFASC(A$)>218THEN1410
1570 IFLEN(P$)=P3GOTO1620
1580 P$=P$+A$
1590 PRINTA$;
1600 GOTO1410
1610 IFLEN(P$)<P4GOTO1410
1620 B$(W)=P$
1630 W=W+1:GOTO1360
READY.
```

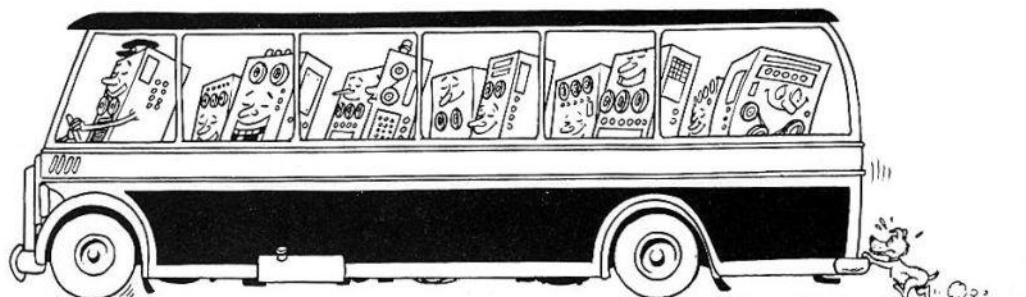
La Redazione di Commodore Computer Club
augura un felice anno nuovo
a tutti i suoi lettori
La memorizzazione dei dati nel VIC
richiede alcuni minuti.
Attendete con pazienza !!!
La memorizzazione dei dati nel Vic 20
richiede alcuni minuti:
Attendete con pazienza !!

Esempi di stampa con la stampante 1515

Modifiche e aggiunte per la versione VIC 20

```
250 OPEN3,4:PRINT#3,CHR$(6):PRINT CHR$(14):POKE36879,8:PRINT"♥"


READY.


290 FORJ=1TO18:READA(J):GOSUB10000: NEXTJ
300 FORH=0TO2:FORJ=1TO6:A$(K,H)=A$(K,H)+CHR$(A(J+6*H)+128):NEXTJ,H,K


READY.


1280 PRINT#3,A$(A,J);:NEXT
1290 PRINT#3,A$(1,1):NEXT


READY.


10000 OP=0:U$="":A$="":U=A(J):FORP=0TO6
10002 IFU/2=INT(U/2)THENU$=U$+"0"::GOTO10004
10003 U$=U$+"1"
10004 U=INT(U/2):NEXT
10010 FORP=7TO1STEP-1:A$=A$+MID$(U$,P,1):NEXT
10011 FORP=1TO7:IFMID$(A$,P,1)="1"THEN OP=OP+2↑(P-1)
10012 NEXT:A(J)=OP:RETURN

READY.
```

C R E A

```
 100 REM ************************
 110 REM **     ' C R E A '    **
 120 REM ************************
 130 REM **PROGRAMMA PER CREARE**
 140 REM **I CARATTERI SPECIALI**
 150 REM **USATI NEL PROGRAMMA **
 160 REM **CORSIVO.            **
 170 REM **VERSIONE PET.       **
 180 REM ************************
 190 REM ** BY BELLU' GIOVANNI **
 200 REM ************************
 210 REM **  TEL. 0362/239580  **
 220 REM ************************
 230 PRINT"♫":DIMA(18)
 240 OPEN5,4,5:OPEN4,4:OPEN6,4,6:PRINT#6,CHR$(25)
 250 INPUT"♫LETTERA:";L$
 260 IFLEN(L$)<>1THEN250
 270 A=ASC(L$)
 280 NL=1000+A
 290 PRINT"♫";
 300 FORK=1TO7:PRINT"┌┬┬┬┬┬┐ ":NEXT
 310 FORK=1TO7:PRINT"×○○○○○◁ ":NEXT
 320 FORK=1TO7:PRINT"└┴┴┴┴┴┘ ":NEXT
 330 INPUTA$
 340 FORK=0TO2:Q=K*7:FORX=0TO5:Z=Z+1:FORY=0TO6
 350 IFPEEK(32768+X+40*Y+40*Q)=81THENA(Z)=A(Z)+2↑(6-Y)
 360 NEXT:NEXT:NEXT
 370 PRINT"♫";
 380 PRINTNL"DATA";:FORK=1TO18:PRINTRIGHT$(STR$(A(K)),LEN(STR$(A(K)))
 390 PRINTCHR$(44);:NEXT:PRINT"▮";CHR$(58)"REM"L$:PRINT"RUN"
 400 PRINT"♫"
 410 FORJ=0TO2:A$="":FORK=1TO6:A$=A$+CHR$(A(K+6*J)):NEXT
 420 PRINT#5,A$:PRINT#4,CHR$(254):NEXT
 430 CLOSE5:CLOSE4
READY.
```

Guida mercato Commodore

Prodotto	Prezzo (IVA esclusa)

VIC - 20

Home Computer Vic 20 199.000
Unità di espansione *(1020)* 295.000
Modulo di espansione *(1023)* 135.000
Cartuccia da 3K di memoria *(1210)* 66.000
Cartuccia da 8K di memoria *(1110)* 98.000
Cartuccia da 16K di memoria *(1111)* 172.000
Cartuccia Vic rel *(4011)* 95.000
Permette di controllare il funzionamento di allarmi antifurto, porte automatiche, telefoni, trasmittenti ed apparecchi similari.
Vic switch *(4012)* 225.000
Possono essere collegati fino a 16 VIC 20 con un floppy e una stampante (distanza massima 1500 mt.).
Interfaccia IEEE 488 *(T-1)* 175.000
Interfaccia centronics *(T-3)* 115.000
RS232-C adapter *(1011-A)* 75.000
RS132-C adapter *(1011-B)* 75.000

Commodore 64

CPU 64K RAM *(CBM64)* 625.000
C 64 Executive *(SX 64)* 2.350.000
Sistema operativo CP/M *(CP/M)* 125.000
Consente di programmare il Commodore 64 in linguaggio CP/M, il più utilizzato sui Personal Computers. Permette inoltre di accedere alla enorme libreria di Software applicativi CP/M.
Pet speed *(6411)* 95.000
Compilatore basic che aumenta la velocità di esecuzione dei programmi di circa 40 volte.

Accessori per Vic e Commodore 64

Stampante plotter a colori *(1520)* 375.000
80 caratteri, per linea, 4 colori, alla risoluzione di 0,2 mm per passo.
Unità stampante *(MPS 801)* 450.000
Stampa velocemente su carta normale quanto appare sul video: programmi, lettere, dati, grafici.
Unità stampante *(1526)* 595.000
Stampante 80 colonne, bidirezionale, 60 CPS, spaziatura programmabile, trazione a frizione o a trattore.
Registratore dedicato *(1530)* 120.000
Per memorizzare facilmente programmi e dati su normali cassette magnetiche.
Floppy disk drive *(1541)* 585.000
Veloce unità di memoria di massa ad alta capacità. Può immagazzinare fino a 170.000 caratteri su ogni singolo disco.
Monitor monocromatico *(1601)* 285.000
A fosfori verdi 12".

Per giocare

Comando per giochi (Joystick) *(1311)* 13.500
Permette di muoversi in tutte le direzioni, di iniziare i vari giochi di movimento e di "sparare".
Comando a manopola per giochi (Paddle) *(1312)* 22.500
Adatto per i giochi a 2 persone, esegue movimenti in orizzontale e verticale,

Commodore 4000

CPU 16K RAM *(CBM 4016)* 1.285.000
18K ROM, BASIC 4.0 residente, video 40 colonne per 25 righe, tastiera semigrafica.
CPU 32K RAM *(CBM 4032)* 1.385.000
18K ROM, BASIC 4.0 residente, video 40 colonne per 25 righe, tastiera semigrafica.

Commodore 9000

Doppia CPU 134K RAM *(CBM 9000)* 2.350.000
Micro Main Frame Computer a doppia CPU (6502 - 6809) compatibile con tutte le periferiche Commodore della serie 8000. Include 5 linguaggi di programmazione. (COBOL, FORTRAN, TCL PASCAL, UCSD PASCAL, APL).

Commodore 8000

CPU 32K RAM *(8032 SK)* 1.675.000
18K ROM, Basic 4.0 residente, video orientabile e basculante 80 colonne per 25 righe, tastiera commerciale separata.
CPU 96K RAM *(8096 SK)* 2.150.000
18K ROM, Basic 4.0 residente, video orientabile e basculante 80 colonne per 25 righe, tastiera commerciale separata. Include sistema operativo PM/96.

Commodore 600

Indicato per applicazioni industriali, collegamento a strumentazione, controllo numerico, ecc. Utilizza monitor in commercio.
CPU 128K RAM *(610)* 2.150.000
CPU 128K RAM espandibile internamente a 256K e esternamente a 960K, interfaccia RS232C, IEEE 488, porta utente a 8 Bit. Compatibile con tutte le periferiche Commodore della serie professionale.
CPU 256K RAM *(620)* 2.550.000
CPU 256K RAM espandibili esternamente a 960K. Caratteristiche uguali al Mod. 610.
Monitor *(1601)* 285.000
Monocromatico a fosfori verdi, 12".

Commodore 700

CPU 128K RAM *(710)* 2.850.000
CPU 128K RAM espandibili internamente a 256K ed esternamente a 960K. Video orientabile e basculante 80 colonne per 25 righe. Compatibile con tutte le periferiche Commodore delle serie professionali.

CPU 256K RAM *(720)* 3.250.000
Monitor a colori 14" con audio. *(1701)* 430.000

Dischi

Floppy disk drive *(2031)* 650.000
Unità di memoria di massa ad alta velocità. Capacità 170KB. Drive singolo.

Floppy disk drive *(4040)* 1.900.000
Unità di memoria di massa ad alta velocità. Capacità 343KB. Drive doppio.

Floppy disk drive *(8050)* 2.125.000
Drive doppio 1M byte in linea.

Floppy disk drive *(8250 H.P.)* 2.450.000
Drive doppio, doppia faccia, doppia densità 2M byte in linea.

Floppy disk drive *(8250 L.P.)* 2.600.000
Drive doppio, doppia faccia, doppia densità, 2M byte in linea.

Hard disk *(9060)* 6.200.000
Tecnologia Winchester, .5M byte in linea.

Hard disk *(9090)* 6.700.000
Tecnologia Winchester, 7.5M byte in linea.

Stampanti

Stampante *(4023 P)* 625.000
Bidirezionale ad aghi, 60 CPS, 80 colonne.

Stampante *(MPP 1.* 1.195.000
Stampante ad aghi 150 CPS, 132 colonne, bidirezionale, trascinamento a trattore.

Stampante *(6400,* 2.950.000
Stampante a margherita, 40 CPS, 136 colonne passo pica, 163 colonne passo élite, bidirezionale, utilizzabile anche con carta da bollo, trascinamento a frizione o a trattore.

Accessori.

Microprocessore 32K RAM *(MUPET II)* 2.500.000
Per connettere, in rete fino a 16 CPU RS232, IEEE 488, centronics. Il prezzo include (configurazione minima): controller, terminator, 3 moduli, cavi, cavo IEEE/PET (per la versione SK).

Singolo modulo addizionale: – 325.000
1 modulo
1 cavo 6 piedi.

Nuovo sistema operativo *(PM 96)* 95.000
Per 8096SK o per 8032SK con B - 1 oppure con B - 2. Può gestire fino a 16 programmi residenti simultaneamente in memoria. Da a disposizione 26K per le variabili e 53K per i programmi. Potenzia inoltre il Basic con altri comandi.

64K RAM *(B-1)* 575.000
Scheda di ampliamento memoria per 8032 e nuovo sistema operativo "PM 96".

CP/Maker *(B-2)* 1.450.000
Incrementa la memoria interna di 64K RAM e permette l'uso di tutti i programmi CP/M. 8 bit disponibili. Compatibile con la serie 3000/4000/8000.

Scheda ad alta risoluzione grafica *(B-3)* 720.000
Compatibile ai sistemi della serie 8000.

Cavo PET/IEEE 488 *(C-1)* 85.000

Cavo IEEE 488/IEEE *(C-2)* 95.000

Accoppiatore acustico *(8010)* .595.000
300 baud/sec.

Ricerca e Tecnologia

XIDEX

presentano il primo dischetto di precisione

- Formulazione ossido ad alta densità lineare
- Coercitività fino a oltre 600 oersted
- Clipphing level al 65%
- Densità fino a 18000 B P I
- Capacità fino a 5 MB
- Disponibili i nuovi dischetti da 3,5"

HC MIL srl
hc magnetic international line
via passeroni 6 - 20135 milano
tel (02) 577477 - 598353 - telex 340216

Comandi per Disco — OPEN X, 8, 15

Comando	Descrizione	Comando	Descrizione
Load "Nome", 8	Carica il prog. Basic: "Nome"	"IØ"	Inizializza il disco
Load "Nome", 8,1	Carica il prog. in Ling. Macc. "Nome"	"VØ"	Cancella i file impropri
Save "Nome", 8	Registra su disco il prog: "Nome"	"CØ: unico =Ø: parte 1,0: parte 2"	Concatena 2 file in un solo "Unico
Save "@Ø: Nome"	Registra, sovrapp., il prog: "Nome"	"CØ: nuovo =Ø: vecchio"	Duplica il "Vecchio" file col nome
Verify " * ",8	Verifica l'ultimo programma	"RØ: nuovo = vecchio"	Cambia in "Nuovo" il nome del "V
"NØ: intestazione, XX"	Formatta il disco con ID = XX	"SØ: primo, secondo"	Cancella i file: "Primo" "Secondo"

LEGENDA	
Comando	Abbrev.
Codice Commodore (decim.)	Codice Commodore (esadec.)

ABS aB 182 B6	AND aN 175 AF	ASC aS 198 C6	ATN aT 193 C1	CHR$ cH 199 C7	CLOSE clO 160 A0	CLR cL 156 9C	CMD cM 157 9D	CONT cO 154 9A
DEF dE 150 96	DIM dI 134 86	END eN 128 80	EXP eX 189 BD	FOR fO 129 81	FRE fR 184 B8	GET gE 161 A1	GOSub goS 141 8D	GOTO gO 137 89
LET lE 136 88	LEFT$ leF 200 C8	LIST lI 155 9B	LOAD lO 147 93	MID$ mI 202 CA	NEXT nE 130 82	NOT nO 168 A8	OPEN oP 159 9F	PEEK pE 194 C2
PRINT ? 153 99	PRINT# pR 152 98	READ rE 135 87	Restore reS 140 8C	Return reT 142 8E	RIGHT$ rI 201 C9	RND rN 187 BB	RUN rU 138 8A	SAVE sA 148 94
SIN sI 191 BF	SPC(sP 166 A6	SQR sQ 186 BA	STEP stE 169 A9	STOP sT 144 90	STR$ stR 196 C4	SYS sY 158 9E	TAB(tA 163 A3	THEN tH 167 A7
VAL vA 197 C5	VERIFY vE 149 95	WAIT wA 146 92	COMANDI SENZA ABBREVIAZIONE			COS 190 BE	FN 165 A5	GET# 161+35 A1+23
INPUT 133 85	INT 181 B5	LEN 195 C3	LOG 188 BC	NEW 162 A2	ON 145 91	OR 176 B0	POS 185 B9	REM 143 8F

Reset del sistema: SYS(x)

VIC 20.X = 64802

CBM64.X = 64738

LEGENDA		
COLORE	CARAT SPEC.	
CHR$	SOLO PER VIC 20	CTRL
SCHERMO		

Aranc. 192 — 8	Marrone 149 — 9 Paglierino	Rosa 150 — 10	Grigio 1 151 — 11 Celeste	Grigio 2 152 — 12 Viola	Verdino 153 — 13	Celeste 154 — 14	Grigio 3 155 — 15 Giallino	Solo CB
Nero 144 — 0	Bianco 5 — 1	Rosso 28 — 2	Azzurro 159 — 9	Viola 156 — 4	Verde 30 — 5	Blu 31 — 6	Giallo 158 — 7	Rev. 18 — CBM 64

VEDI CHR$ (8) CHR$ (9)

CHR$ (14) CHR$ (142)

PUNTATORI PIU' USATI		
LO	HI	RELATIVI A:
43	44	START BASIC
45	46	START VARIABILI
47	48	START ARRAY
49	50	FINE ARRAY
51	52	STRINGHE
53	54	FINE STRINGHE
55	56	TOP OF MEMORY

SUONO COMMODORE 64 — **LOCAZIONI UTILI**

Locazione		Valori
54272 / 54273	NOTE MUSICALI	Ø ÷ 255
54276 FORME D'ONDA	17 = TRIANG. 33 = SINUSOI. 64 = PULSAN. 129 = RUMORE	
54274 / 54275	SOLO PER ONDE PULS.	Ø ÷ 255 / Ø ÷ 15

	BIT 4 ÷ 7	BIT Ø ÷ 3	
54277	ATTACK	DECAY	Ø ÷ 255
54278	SUSTAIN	RELEASE	Ø ÷ 255

54296 VOLUME Ø ÷ 15

N.B. X = PRIMA VOCE
X + 7 = SECONDA VOCE
X + 14 = TERZA VOCE

SUONO VIC 20 (PO

X = 36874.Y = 128 ÷ 255
X = 36875.Y = 128 ÷ 255
X = 36876.Y = 128 ÷ 255
X = 36877.Y = 128 ÷ 255
X = 36878.Y = Ø ÷ 15

PRINT CHR$ (X)

5 Bianco	37 %	55 7	73 I	91 [	109	127	158 Giallo	176	184	129 Aranc.
8 Ca. Blocco	38 &	56 8	74 J	92 £	110	133	159 Azzur.	177	185	149 Marr.
9 Ca. Sblocco	39 '	57 9	75 K	93]	111	134	160 Spazio	178	186	150 Rosa
13 Return	40 (	58 :	76 L	94 ↑	112	135	161	179	187	151 Grigio 1
14 Min-Mai	41)	59 ;	77 M	95 ←	113	136	162	180	188	152 Grigio 2
17 CRSR↓	42 *	60 <	78 N	96	114	137	163	181	189	153 Verdino
18 RVS.ON	43 +	61 =	79 O	97	115	138	164	182	190	154 Celeste
19 Home	44 ,	62 >	80 P	98	116	139	165	183	191	155 Grigio 3
20 Del	45 –	63 ?	81 Q	99	117	140	166			
28 Rosso	46 .	64 @	82 R	100	118	141 Shift+ Return	167			
29 CRSR→	47 /	65 A	83 S	101	119	142 Maius.	168			
30 Verde	48 0	66 B	84 T	102	120	144 Nero	169			
31 Blu	49 1	67 C	85 U	103	121	145 Curs.↑	170			
32 Spazio	50 2	68 D	86 V	104	122	146 RvsOff	171			
33 !	51 3	69 E	87 W	105	123	147 Home	172			
34 "	52 4	70 F	88 X	106	124	148 Ins.	173			
35 #	53 5	71 G	89 Y	107	125	156 Viola	174			
36 $	54 6	72 H	90 Z	108	126 π	157 ← CRSR	175			

192 ÷ 223 Vedi 96 ÷ 127
224 ÷ 254 Vedi 160 ÷ 190
255 = 126

LEGENDA: CHR$
- Codice Ascii
- Caratteri Grafici
- Colori aggiuntivi CBM 64
- Comandi speciali

lo "Unico"
col nome "Nuovo"
me del "Vecchio" file
"Secondo"

cO 9A	DATA 131	dA 83
gO 89	INPUT# 132	iN 84
pE C2	POKE 151	pO 97
sA 94	SGN 180	sG B4
tH A7	USR 183	uS B7
# 5 A1+23	IF 139	8B
8F	TAN 192	C0

Solo CBM 64

CBM 64 e VIC 20

Rev. Off. 146

commodore COMPUTER CLUB

NORM + SHIFT

Cancella schermo 147	Inserim. caratt. 148
Cursore in alto 19	Cancella caratt. 20

Norm + Shift

F1 133	F2 137
F3 134	F4 138
F5 135	F6 139
F7 136	F8 140

A. DE SIMONE

NORM + SHIFT

CURSOR ↑ 145	CURSOR ← 157
CURSOR ↓ 17	CURSOR → 29

VIC 20 (POKE X, Y)

128 ÷ 255 Prima tonalità
128 ÷ 255 Seconda tonalità
128 ÷ 255 Terza tonalità
128 ÷ 255 Rumore bianco
Ø ÷ 15 Volume

COLORI	BORDO	FONDO	Val. Standard
VIC 20	36879 BIT Ø ÷ 3	36879 BIT 4 ÷ 7	27
CBM 64	53280	53281	254 246

VIC 20		Versione base	+ 3K RAM	+8 ÷ 16 K
Basic	#	4096-7679	1024-7679	4608 ÷
	$	1000 1DFF	0400-1DFF	1200 ÷
Schermo	#	7680-8191	7680-8191	4096-4607
	$	1E00-1FFF	1E00-1FFF	1000-11FF
Colore	#	38400-38911	38400-38911	37888-38399
	$	9600-97FF	9600-97FF	9400-95FF

CBM 64	BASIC	SCHERMO	COLORE
#	2048-40959	1024-2023	55296-56295
$	0800-9FFF	0400-07E7	D800

La maschera - memo del programmatore

Come usare l'inserto speciale offerto in omaggio da Commodore Computer Club

Servendosi di un tagliabasa ritagliare, con grande attenzione, la parte interna dell'inserto (zona 0). I più esigenti verificheranno, prima del taglio, l'esatta dimensione della tastiera, in modo che la coperta-inserto si adatti perfettamente sul Vic 20 o sul Commodore 64. I tagli da effettuare, seguendo le rispettive sagome, sono tre: quello relativo alla tastiera Qwerty, quello relativo ai quattro tasti posizionati a destra e quello della lampadina-spia.

Interpretazione della tabella

• *Zona 1.*
Pro-memoria per il colloquio con i dischi. Dal sesto rigo in poi è ovviamente sottintesa una precedente apertura del colloquio (OPEN#... ecc.).

• *Zona 2.*
Istruzioni e comandi del Basic Commodore. In ciascun rettangolino sono inserite quattro informazioni:
a) in alto a sinistra il comando nella sua dizione ortodossa così come deve essere battuto nel caso in cui si sia nell'opzione caratteri maiuscoli-grafici;
b) in alto a destra lo stesso comando in forma abbreviata indicato, per facilitarne la comprensione, nella forma in cui appare se si imposta l'opzione caratteri minuscoli-maiuscoli. Esempio: invece di digitare RUN si può battere il tasto "r" ed il tasto "U" (u + shift);
c) in basso a sinistra il codice decimale interno Commodore;
d) in basso a destra lo stesso codice espresso in esadecimale. Queste due ultime informazioni interessano gli utenti "evoluti" che hanno spesso occasione di leggere il contenuto della memoria del computer.

• *Zona 3.*
Reset dei sistemi. Digitando la SYS relativa e poi il tasto Return (R), viene simulato lo spegnimento e la riaccensione del computer.

• *Zona 4.*
Codici colore. In ciascun rettangolino sono presenti più informazioni:
a) sulla destra è raffigurato il carattere speciale che appare nei listati nei casi di impostazione dei colori;
b) in alto sulla sinistra sono indicati i colori ottenibili;
c) in basso a sinistra il valore decimale che usato come argomento in CHR$(X) setta il colore che deve essere usato. Esempio:

PRINT CHR$(5)

colora in bianco i caratteri che saranno successivamente stampati sul video;
d) in basso a destra il codice colore da usare nel comando POKE, X in cui X rappresenta l'indirizzo di una cella video. Si tenga presente che la riga alta della zona 4 contiene valori ottenibili direttamente da tastiera solo col Commodore 64, mentre quella inferiore si riferisce ai colori ottenibili premendo il tasto CTRL insieme ad uno dei tasti-colore.

• *Zona 5.*
Codici ASCII e Commodore. Sono i caratteri o i comandi speciali ottenibili con:

PRINT CHR$(X)

Esempio: PRINT CHR$(87) fa apparire il carattere W.

PRINT CHR$(19) sposta il cursore in alto a sinistra, ecc.

• *Zona 6.*
Informazioni utili sulla mappa della memoria relativa allo schermo e al basic.

• *Altre zone.*
Data la loro semplicità di interpretazione evitiamo di commentarle.

Per rendere più "professionale" le tabelle, consigliamo di incollare con cura l'inserto, servendosi di nastro biadesivo, su di un foglio sottile di compensato o di plexiglas per conferire rigidità. La dispersione del calore non dovrebbe essere in alcun modo compromessa. La Rivista declina comunque ogni responsabilità derivante dall'uso dell'inserto che, assicuriamo, è stato esaminato con gran-cura prima della pubblicazione.